REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* April 23, 2004	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From - To)* April 2001 to April 2004

4. TITLE AND SUBTITLE (Do not use all capital letters)
Understanding and Applying the Cognitive Foundations of Effective Teamwork

5a. CONTRACT NUMBER N00014-01-C-0347

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Dr. David F. Noble

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
EVIDENCE BASED RESEARCH, INC.
1595 Spring Hill Road, Suite 250
Vienna, VA 22182-2216

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ballston Tower One
800 North Quincy Street
Arlington, Virginia 22217

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT (200 word max)
Cognitive issues, though not as widely discussed as resource or social issues, are key to the success of collaborating teams. This report describes important cognitive issues. It presents several examples of team failure caused by cognitive problems. It reviews a theory describing the knowledge that teams need to work together effectively and summarizing how teams use this knowledge when making decisions about collecting and sharing information, attaining consensus, and adapting the team. It also describes an expert system, the Collaboration Advizor tool, that helps teams diagnose and fix cognitive-based collaboration problems.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF: UNCLASSIFIED			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** David F. Noble
a. REPORT UNCLASSIFIED	**b. ABSTRACT** UNCLASSIFIED	**c. THIS PAGE** UNCLASSIFIED		91	**19b. TELEPHONE NUMBER** *(include area code)* (703) 287-0312

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

UNDERSTANDING AND APPLYING THE COGNITIVE FOUNDATIONS OF EFFECTIVE TEAMWORK

PREPARED BY
Dr. David F. Noble

PREPARED FOR
Office of Naval Research

Human Sciences Department

Dr. Michael Letsky, COTR

Contract N00014-01-0347

DATE
April 23, 2004

Executive Summary

This report summarizes some of the work performed under an Office of Naval Research SBIR, Contract N00014-01-C-0347, from April 2000 through March 2004. It describes two major accomplishments of this work, development of a cognitive theory of collaboration describing the roles and types of knowledge important to collaboration, and development of the Collaboration Advizor™ Tool, an expert system that helps teams diagnose and fix knowledge-related collaboration problems. In addition to the work described in this report, other publications have described how the theory can support development of collaboration metrics (Noble and Kirzl, 2003; Kirzl et al, 2003), help partition team functions among human and computer team members (Noble, 2003a, Noble 2003b), guide selection of collaboration tools (Noble, 2004a), or help manage teams (Noble, 2004b).

Importance of Knowledge to Effective Teamwork

In today's network-connected world, collaboration and teamwork are becoming increasingly important. Effective collaboration, both for co-located and spatially distributed teams, can improve the quality and timeliness of assessments and decisions. Effective teams can flawlessly synchronize, quickly adapting themselves to seize opportunities and thwart risks.

Unfortunately, collaborating teams are not always effective. They do not always identify viable alternatives for action, do not always reach a good understanding of situation events, and do not always generate high quality decisions. A team's failure is always undesirable, and sometimes can have disastrous consequences.

Teams can fail for one of three basic reasons: inadequate resources, lack of knowledge of how to do needed tasks or work together as a team, or unwillingness to do the work. Much has been written about the first and third causes of failure. There has been much less discussion about the second cause, cognitive shortfalls. Yet, cognitive problems can be a team's undoing. A famous example of this was the Kennedy administration's Bay of Pigs fiasco. This operation attempted "to place a small brigade of Cuban exiles secretly on a beachhead in Cuba with the ultimate aim of overthrowing the government of Fidel Castro" (Janis, 1972). The operation was an immediate military failure with all of the participants being either killed or led to prison camps.

This team did not fail because of inadequate resources or poor social interactions. This was a high motivated, extremely capable and knowledgeable cabinet level group that could draw on the full capabilities of the federal government. Instead, this team failed for cognitive reasons. The Kennedy team did not adequately focus on the full range of possible outcomes and did not anticipate some of the more harmful long-term effects of its actions. Team members were not aware of critical information that they needed to consider. Poor team business rules led to "poor sharing of information, unwillingness to share private information, suppression of personal doubts, unwillingness to obtain outside information to test assumptions," and a "taboo about antagonizing valuable new members."

As this and the additional examples of Section 1 point out, up-to-date problem solving tools or the brightest team members do not ensure team success. Rather, in order to promote success (and sometimes avert disaster) team members need to understand the cognitive basis of effective teamwork.

The Human Systems Department of the Office of Naval Research (ONR) has supported research to understand the cognitive basis of team effectiveness. The work described in this report is one part of ONR's program in the cognitive foundations of effective teamwork and collaboration. This work on the role of knowledge in collaboration contributes to an improved understanding of the importance and nature of knowledge building in collaboration and teamwork. It has helped to de-mystify the mechanics of teamwork and to move its management a little more from just being an art to that of a science. It has also led to the development of a team knowledge diagnostic tool that helps people use this more codified understanding of important team knowledge.

A Theory of Knowledge in Collaboration

To help teams deal with cognitive issues in teamwork, this work synthesized a theory describing the role of knowledge in collaboration, and then, building on that theory, developed the "Collaboration Advizor™" expert system that helps teams diagnose and fix cognitive-related collaboration problems. The theory, detailed in Section 2 of this report, describes the importance of knowledge to effective collaboration and discusses general categories of the knowledge teams need to interpret and share information, to achieve consensus, and to determine when the team needs to modify how it works.

The heart of Section 2 is a detailed description of exactly what knowledge team members need in order to work together effectively. Developing this detailed description presented numerous challenges, for collaboration depends on many different types of knowledge documented and described in diverse literatures on collaboration, teamwork, project management, decision making, situation understanding, and command and control.

Because it is sometimes difficult for teams to be aware of everything they need to know, the ONR research team organized the needed knowledge into twelve "knowledge enablers," describing for each exactly what needs to be known, why and when this knowledge is important, how to get it, and how to tell when the team lacks it.

The Twelve Knowledge Enablers

1. Goals: understanding requirements and what a good result looks like.

2. Roles, tasks, and schedule: includes knowing progress indicators

3. Relationships and dependencies: understanding how information and resources impact tasks and how tasks impact goals.

4. Team members' backgrounds and capabilities: understanding what others can and will do under various circumstances. Key to trust.

5. Team "business rules:" knowing the agreed-upon procedures for information, helping each other, and resolving conflict.

6. Task knowledge: knowing how to do your job.

7. Activity awareness: knowing what others are doing, how busy they are, and whether they are working on the right thing.

8. The external situation: understanding adversaries and competitors and how they can impact team success

9. Task assessment: tracking task progress

10. Mutual Understanding: knowing the extent to which team members agree or disagree.

11. Plan assessment: understanding whether the team's plan will still work

12. Decision drivers: understanding decision drivers, deadlines, handling uncertainty, and who to consult

The Collaboration Advizor™ Tool

To help teams apply this understanding about needed knowledge, ONR funded Evidence Based Research, Inc (EBR), based in Vienna, Virginia to encode this knowledge into an expert system tool that helps people to diagnose and fix problems. This tool helps team members understand the knowledge critical to team effectiveness, helps them recognize and discuss areas where the team may be having knowledge difficulties, helps them surface and discuss areas of disagreement, and helps them identify ways to improve team knowledge and performance. With this tool, teams can catch small knowledge problems early, before they turn into big issues.

Section 3 describes this tool. It describes how teams use the tool as part of their team life cycle. It describes how the tool works, discussing its knowledge base and a few of the most critical algorithms. Finally, it reviews the tool's evolution and the evidence indicating its effectiveness and utility.

Table of Contents

List of Figures

1.0 Introduction

1.1 Benefits and Challenges

In recent years, collaboration has become increasingly important. Improved communications infrastructure, such as peer-to-peer collaborative information delivery, is now widespread, increasing the feasibility of distributed collaboration. Industry and government increasingly understand the benefits of collaboration wherein team members leverage each other's knowledge and experience to generate more reliable and higher quality assessments, plans, and decisions often faster than any one individual could produce working alone.[1]

Aware of these benefits, industry, government, and academia are examining different team organizations such as self-managed teams that depend on the cognitive glue stemming from common team member understandings. The new military doctrines of Network Centric Warfare (NCW) and the Navy's ForceNet build on networking and the collaboration it supports. Multicultural collaboration is becoming more common and more important. The success of recent military operations such as Operations Enduring Freedom in Afghanistan and Iraqi Freedom in Iraq were significantly helped by the contributions of collaborating coalition partner's and non-governmental organizations.

Unfortunately, collaborating teams often need to work in tough environments under difficult conditions that can impede effective teamwork. For example, teams may draw their members from diverse disciplines, backgrounds, and cultures. They may be spatially distributed, and work in different time zones and different shifts. They may be presented with poorly structured and vaguely formulated problems, and often need to handle large amounts of unstructured and uncertain information. The nature of these problems may preclude relying on fixed and clear processes for teamwork. Instead, team members may need to seek out and distribute information in many different ways, depending on the circumstances.

To illustrate the challenges of effective teamwork, this introduction reviews four examples of team and collaboration failures. The first two, the Bay of Pigs and the Vincennes shoot down of the Iranian airliner, led to tragic results. Both have been extensively studied. Fortunately, these catastrophic failures are infrequent. Most team and collaboration failures just lead to increased work and disappointing results. To illustrate these smaller but much more frequent setbacks, examples 3 and 4 describe two of the twenty case studies of collaboration failures contributed by our sub-contractor Klein Associates and used in formulating our cognitive framework of collaboration.

Example 1: The Bay of Pigs

Irving Janis, in his classic analysis of groupthink[2] characterized the Kennedy administration's Bay of Pigs action as "an operation so ill conceived that among literate people all over the world the name of the invasion site has become the very symbol of a

[1] Evidence Based Research, 2001.
[2] Janis, 1972.

perfect failure."[3] This disaster occurred even though in many ways, the policy team responsible for the Bay of Pigs did not have any of the features that would normally be expected to impede team performance. It consisted of extremely talented and knowledge people. Team size was small, fewer than fifteen people. The team was collocated; everyone could easily meet face to face. And the team was highly motivated. The team achieved consensus, but this Bay of Pigs consensus position led to disaster, illustrating that even intelligent and highly motivated teams working in an ideal physical environment are not always effective.

The Bay of Pigs operation attempted "to place a small brigade of Cuban exiles secretly on a beachhead in Cuba with the ultimate aim of overthrowing the government of Fidel Castro." The operation was an immediate military failure. "Nothing went as planned. On the first day, not one of the four ships containing reserve ammunition and supplies arrived… By the second day, the brigade was completely surrounded by twenty thousand troops of Castro's well-equipped army. By the third day, about twelve hundred members of the brigade, comprising almost all who had not been killed, were captured and ignominiously led off to prison camps."

This immediate failure was followed by highly deleterious, unanticipated long-term effects. The "U.S. government's attempt to disclaim responsibility for the initial air assault" was "thoroughly discredited, friendly Latin American countries outraged, world wide protests" denounced the government for its "illegal acts of aggression against a tiny neighbor." And most seriously, the operation encouraged "military rapprochement between Castro and the Soviet leaders, culminating in a deal to set up installations only ninety miles from United States' shores equipped with nuclear bombs and missiles and manned by more than five thousand Soviet troops. This transformed Cuba within eighteen months into a powerful military base and a satellite of the Soviet Union."

Example 2: The Iranian airliner shoot down (Klein, 1998)

On July 3, 1988, the U.S. Aegis Cruiser Vincennes mistakenly shot down an Iranian commercial airliner. This occurred during the Iraqi-Iranian conflict. An Iraqi fighter had recently attacked the USS Stark, killing several of the crew. Earlier on July 3, Iranian gunboats had attacked the USS Elmer Montgomery.

The Iranian airliner departed from Iran at 10:17 Iranian time, 27 minutes after its scheduled 9:50 departure time. Seven minutes and eight seconds later, the Vincennes destroyed the airliner. What occurred in the interval was a series of unfortunate coincidences and misunderstandings leading Captain Will Rogers to believe that the unknown track was an Iranian fighter intending to attack the ship. Principal indicators of hostility were (1) the aircraft took off from a dual use airport that military aircraft also used, and so was designated as "unknown, presumed enemy"; (2) there was no commercial flight scheduled to take off at 10:17 (the aircraft departed 27 minutes late), there was an Iranian P3capable of providing targeting information in the area; (3) the track was heading toward the Vincennes (unfortunate, but in accordance with its flight

[3] Ibid, p. 9.

plan); (4) the airliner pilot never responded to the Vincennes challenges and requests for information; (5) several crew members (wrongly) reported that the track's IFF designated it as an non-U.S. military aircraft; (6) the aircraft was (wrongly) reported as descending, a profile inconsistent with a commercial airliner but expected were it preparing to attack the Vincennes.

As noted, the Vincennes's conclusions were incorrect. In fact, they were unnecessarily incorrect. Another nearby U.S. ship, the USS Sides, was also tracking the track and correctly classified it as an airliner. The investigations that followed the incident concluded that all the information needed for a correct decision was either somewhere on the ship or could have been obtained; but it just was not assembled and interpreted properly. For example, the ship could have called the airport and established that a commercial airliner took off at 10:17. The correct information that the track was actually ascending was available. It was just hard to see, because the ship's displays showed only altitude and not change of altitude. The incorrect IFF seemed to be caused by careless information handling. The request associated the IFF with a place rather than with the track. The IFF interrogation lasted 90 seconds, longer than it should have. Possibly in that time, a military aircraft did go to that position. In addition, there were several counter-indicators of an attack. The track never emitted the radars needed for an attack and it did have the IFF response appropriate for an airliner.

Example 3: Performing work no longer needed and too late to be useful

In a training exercise, a fire was reported in a room that contained an electrical box. By the time the engineering group was alerted, the simulated fire had already been extinguished with no harm to the electrical system. However, no one informed the engineers of this, and they spent the next hour and a half planning no longer needed contingencies should the electrical system be damaged. After they completed their plans and were about to present them to the entire Technical Support Center, they discovered that the fire was out. If the fire had been real, their planning would have come far too slowly to be useful. Yet they were congratulated on the quality of their plans.

In this example a team's not all being aware of its tasks and how they fit together led to two problems. First, a sub-team performed unneeded work. It was needed when they started, but then was continued even though the requirement for doing the work had disappeared. Second, even if the work had been needed, the sub-team performed it far too slowly to have met the needs of the team.

Interestingly, in the training debrief to the team, it became apparent that the team had not been aware that they had not performed well. No one realized that they should have notified the engineers that the fire was out before they even began their deliberations. No one noted that the contingencies would have been presented too late to be useful. The problem was not that no one knew the situation had changed. Though the main group did monitor the situation and saw the need for the fire plan was overtaken by events, they did not realize that they needed to adjust the sub-team's goal.

Example 4: Taking the Wrong Action

In a small company, one individual had the job of keeping the computer systems working. One day a mouse stopped working that was attached to one of the main (critical) servers. The systems operator (sysop) sent a request to purchase a replacement. To make sure the request was perfectly clear, he tracked down when the original mouse had been ordered and wrote that he wanted the exact same mouse. He even included the date of the earlier purchase order for reference. He believed he had covered all bases. He had done a careful and thorough job. There should be no ambiguity.

To the sysop's surprise, the replacement mouse didn't work; it didn't even fit. Somehow the front office has ordered the wrong one. In tracking down the reason, the sysop found that the hardware company no longer made the original mouse.

The front office had assumed that the sysop was trying to indicate the company he preferred to order from. They contacted that company and ordered the mouse closest in price to the original. They were not aware that there was a compatibility problem, that not all mice fit all machines. To complicate matters further, the sysop was traveling when they ordered the mouse, so they could not ask him. Since they felt that he wanted the replacement quickly, they did not wait for him to return. They wanted to show how responsive they were.

1.2 The Cognitive Causes of Team Problems

There are three basic reasons why teams fail: problem difficulty, knowledge, and will.

- Sometimes teams fail because the problem is too hard, the environment too tough, the organization too unwieldy, the team members are not smart enough, or the resources too limited for any team to succeed. Were a neighborhood to create a local youth basketball team with the goal of beating the Los Angeles Lakers at the end of the season, they would fail, as would any such team. They would not have the talent and resources.

- Sometimes teams fail because they do not have the knowledge they need. They have the resources, but lack the experience and know-how to solve a problem. A team of novices who have never prepared proposals in the past and have little experience in the domain being proposed is unlikely to write a winning proposal despite a generous budget and extended preparation time. They just do not know how to write this proposal. These are the cognitive causes of failure.

- Sometimes teams fail because of poor motivation. They have the resources and knowledge, but just do not care enough about succeeding to do the necessary work. Teams who feel that their managers treat them badly and will not reward or recognize success may let a project fail rather than put in the extra overtime needed for success. These are the motivational causes of failure.

Most advice books on team success address resource and motivation causes of failure. For example, planning guides describe how to determine the resources that one’s team needs to succeed. Popular programs in team building, such as Outward Bound, address motivation, helping people get to know, care about, and trust each other.

However, many teams fail because they do not have the knowledge they need to succeed. They do not have the knowledge to do their tasks, and they do not know how to work together effectively. Their errors are cognitive. This was the case in each of the four preceding examples.

The Kennedy team approving the Bay of Pigs operation did not fail for lack of resources, for this cabinet-level group could draw on the full capabilities of the federal government. It did not fail for organizational reasons, for team size was small with fewer than ten people, and was collocated, permitting everyone to easily meet face to face. It did not fail because the team members were not smart enough, for the team members were extremely capable. Nor did it fail because the team did not try hard enough, for everyone on the team was highly motivated.

Most analyses of the Vincennes case have concluded that the information needed for a correct decision was on the ship. Another ship with the same sources of information understood the situation correctly. On the Vincennes, however, this information was not assembled and analyzed correctly. The ship had the resources, and all personnel were highly motivated to take the right action. The failure was cognitive.

The engineers in the training exercise continued their no-longer-needed and tardy planning because they were not aware that the danger from the fire had passed and because they did not know how the team’s tasks fit together. The cause of the problem was cognitive: poor situation awareness, poor knowledge of update procedures, and poor understanding of task deadlines.

The confusion over ordering the right mouse also occurred for cognitive reasons. The purchaser lacked “domain” knowledge. He didn’t know that there were different types of mice and that he needed to check back with the requester. The requester did not have a good understanding of the purchaser’s knowledge, and so did not specify the required type of mouse. This type of “transfer of meaning” error is a common cognitive problem in teams.

1.3 Theories and Tools to Improve Team Cognitive Performance

The lesson from these examples is that success depends on more than just outfitting the team with the most up-to-date problem solving tools or recruiting the brightest team members. Nor is it enough to ensure that all team members are highly motivated and dedicated. Rather, good teamwork also depends on getting the cognitive factors right. Getting these right encompasses not only knowing what teams need to know, but also includes knowing when that knowledge is needed, what makes it hard to obtain, what happens when it is missing, how to tell when it is missing, and what to do after you determine that it is missing.

Unfortunately, knowing how to recognize and fix cognitive team problems remains a mysterious art. It is not always clear how to fix teams with such problems. Until recently, the only resources available to leaders working with teams have been popular "managing for dummies" type books and a few niche consultants specializing in team development. Although this support at times proves to be helpful, it often varies in quality and is largely drawn from intuition and experience. Moreover, it is rarely built upon a scientific foundation, and focuses primarily on only two of the three major causes of team failure: inadequate resources and poor social interactions.

The Human Systems Department of the Office of Naval Research (ONR) has supported research to address the cognitive issues important to effective team performance. This research helps identify the knowledge needed to work together effectively. Although it is obvious that teams that do not have the knowledge needed to work together well are unlikely to do so, it has not been easy to figure out how to act when such a situation exists. That is, it has not been easy to pin down exactly what this needed knowledge is, to know the circumstances when particular types of knowledge are especially important or difficult to obtain, to recognize when a knowledge gap is hindering team performance, and then to know how to fill the gap. The problem has been that the required knowledge has not been cataloged and linked to diagnostic tests and proven remedies in a systematic way.

The work described in this report, one part of ONR's program in the cognitive foundations of effective teamwork and collaboration, has addressed these issues, leading to a greater understanding of the importance and nature of knowledge building in collaboration and teamwork. It has helped to demystify the mechanics of teamwork and to move its management a little more from just being an art to that of a science. It has also led to the development of a team knowledge diagnostic tool that helps people organize and use this more codified understanding of important team knowledge.

To help teams deal with cognitive issues in teamwork, this work synthesized a theory describing the role of knowledge in collaboration, and then, building on that theory, developed a collaboration advisor expert system that helps teams diagnose and fix cognitive-related collaboration problems. Developing a theory explaining the role of knowledge in collaboration presented numerous challenges, for collaboration depends on many different types of knowledge documented and described in diverse literatures on collaboration, teamwork, project management, decisionmaking, situation understanding, and command and control. Building a practical tool that helps teams take advantage of this theory presented additional challenges, for the tool needed to provide its guidance in a straightforward, easy to understand, and easy to use format.

1.4 Overview of report

This report has two additional sections. Section 2, "How Teams Work," describes the theory. Section 3, the Collaboration Advizor™ Tool, documents the expert system that helps teams to apply the theory.

Section 2 begins by describing collaboration and teamwork. It discusses the different types of teams, what teams do, and metrics for measuring how well a team is doing. It then details the core of the theory—the knowledge that team members need in order to work together effectively, and the organization of this knowledge into twelve "knowledge enablers." It describes for each of these enablers the circumstances under which having that knowledge is most important, the consequences of gaps or deficiencies in that knowledge, ways to determine if the team has deficiencies in this knowledge, and remedies to correct these deficiencies.

Though the center piece of the theory is the set of twelve enablers that describe critical team knowledge, Section 2 also briefly addresses the cognitive processes that team members use to acquire and share this knowledge and to reach consensus. It discusses a default reasoning model that describes cognitive mechanisms individuals may use when they integrate new information to update current beliefs and make inferences and projections. This default reasoning model helps explain the "hardening" that occurs as team members work together over time. It also helps explain the cognitive basis of team difficulties from multicultural team members, new members, and from cognitive biases. In addition, Section 2 discusses the evaluative "meta-knowledge" that people use to make decisions about acquiring and sharing information and achieving consensus and the "opportunistic reasoning" that helps team quickly adapt to new circumstances.

Section 3 describes the Collaboration Advizor™ Tool that helps teams address cognitive issues and perform more effectively. It describes the physical parts and look and feel of the tool. It discusses how teams would use the tool as part of their team development life cycle. It also briefly reviews how the algorithms and knowledge base organization that the tool uses to make its diagnoses and recommendations. It concludes by reviewing the development cycles and validation results.

2.0 How Teams Work—A Cognitive Theory of Teamwork and Collaboration

2.1 Definitions of Teamwork and Some Metrics for Measuring Team Performance

2.1.1 Definitions

The theory described in this report applies to all kinds of teams, not only the "thinking" teams that are the principal focus of this work, but also the action teams that execute military plans, ensure customer service, and win basketball games. As proposed by Eduardo Salas(1992), a team in general can be defined as

> *"a distinguishable set of two or more people who interact, dynamically, interdependently, and adaptively toward a common and valued goal/ objective/mission, who have each been assigned specific roles or functions to perform and who have a limited life-space of members."*[4]

"Thinking" teams create intellectual products collaboratively. ONR has characterized such collaboration as

> *"the mental aspects of joint problem solving for the purpose of achieving a shared understanding, making a decision, or creating a product."*

Others have emphasized the importance of people leveraging each others' expertise to create a product that no one team member could have produced alone. This focus characterizes collaboration to be

> *experts integrating perspectives to better interpret the situation and problem, identify candidate actions, formulate evaluation criteria, and decide what to do.*

In keeping with this view of collaboration, one can characterize the benefits of collaboration as enabling collaborating teams to "make better lists." For instance, in contrast to a single person planning alone, a collaborating planning team will generate:

- A better set of views on what is happening, the reasons for these occurrences, and their impacts on the team mission;
- A better set of candidate actions to take in response to these impacts;
- A better set of criteria to consider when evaluating the desirability of these actions; and
- Better estimates of possible consequences of the alternatives being considered.

[4] Salas et al., 1992 as quoted in Mathieu, 2000.

Like thinking teams, *action teams* also work together to create a result that no one on the team could create alone. However, unlike thinking teams which leverage each others cognitive processes, members of action teams leverage each others' physical processes. Thus, an action team can cover more territory than can any single member, can share workload, can be active 7 days a week and 24 hours a day, and can augment each others' special skills so that the team can handle a greater number of situations.

2.1.2 A Team Taxonomy

Thinking and action teams are the beginning of a taxonomy. However, they are only a start to capturing the full range of the large variety of types of teams. In Phase 1 of this SBIR, our subcontractor Klein Associates suggested the much more complete and insightful taxonomy presented in Table 1.[5] This more detailed taxonomy is important in understanding the role of knowledge in collaborating teams, for different kinds of knowledge are more or less important in different kinds of teams.

Table 1. Taxonomy of Collaboration Teams

Team Dimension	Dimension Subcategories
Distribution	• Physical—spatial separation of team members • Temporal—e.g., working different shifts • Expertise—spatial and temporal distribution of experts and expertise • Information—spatial distribution of information
Roles and Functions	• Stability of definition—whether roles are clearly defined or become defined in the process of performing work • Experience—extent that each team member is experienced with assigned roles and collaboration tools • Familiarity—extent each team member is familiar with roles and functions of other team members • Team member expertise—extent that individual team members have specialized expertise needed for their assigned tasks
Team Structure	• Hierarchical vs. flat—extent that team has designated leader in charge or is peer-to-peer • Size—number of members • Permanent vs. ad hoc—extent it works together over extended period of time, or is brought together for one task • Single vs. team-of-teams—extent that teams can be decomposed into collaborating sub-teams • Turn-over—stability of team membership
Information and Information Flow	• Information sharing—degree to which team members need to share information • Information processing complexity—number of hand-offs required to produce an information product • Team expertise—extent that expertise the team needs as a

[5] Noble, 2000.

	whole resides somewhere within the team or that team members must go outside of team to retrieve needed expertise
Team Dimension	**Dimension Subcategories**
Team member dependencies	• Independence—extent that each team member depends on other team members to perform his task • Interaction frequency—how often team members must interact • Synchronization—requirement for and schedule tolerance of temporal sequencing of tasks performed by different members • Cognitive—extent that team members must pay attention to each others' tasks • Task sharing—extent to which each team member has own task or all team members share the same tasks • Processing flow—individual/parallel or sequential
Decisionmaking	• Group makes decision vs. leader makes decision • Reactive vs. proactive—extent that tasks require team to react to uncontrollable events • Degree of time pressure • Stakes—degree of risk and responsibility

This taxonomy of collaboration teams describes a small number of major dimensions and a larger number of dimension subcategories by which teams may be classified. Because these dimensions are mostly independent, the values that a team takes along one dimension normally do not constrain the values that the team may take along other dimensions.[6]

To apply this taxonomy, each dimension listed in Table 1 must be associated with a set of possible dimension values. For example, the values of physical spatial separation may be "co-located within speaking distance," "co-located within the same building," "located within an hour's drive," "or geographically dispersed and difficult to arrange face to face meetings."

The difficulty of teamwork can be strongly affected by the factors listed in Table 1. For example, it is harder for people to work together if they are spatially separated than if they are co-located, or if they work in different shifts rather than in a single one. It is harder for teams to succeed if roles are not clearly defined, if the members of the team keep changing, if the team members' work is highly interdependent, if information must be handed off several times in order to produce a product, and if tasks require the team to react to uncontrollable events.

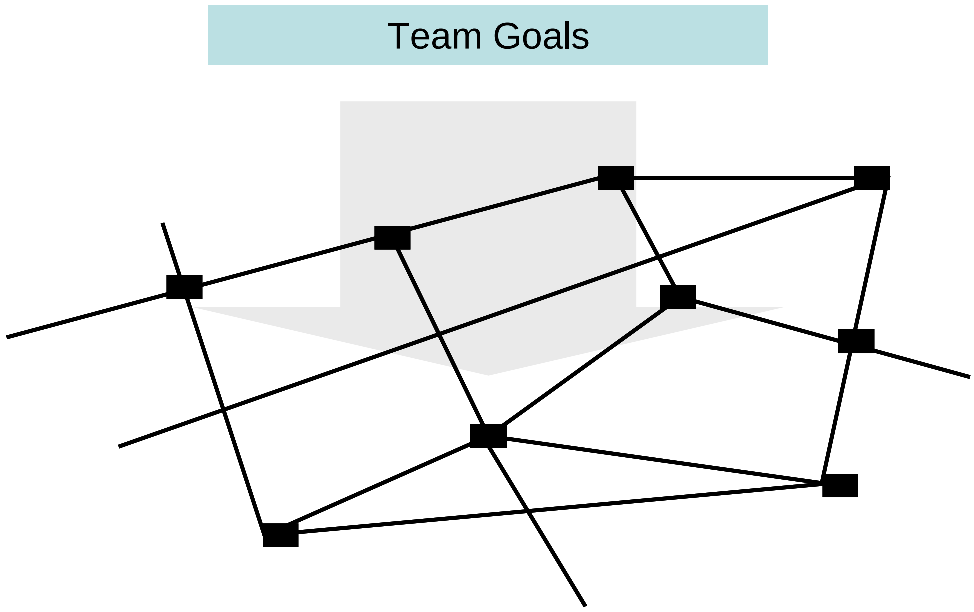
Figure 1[7] illustrates a self-managed team of peers, a type of team that is becoming of increasing interest. In this team, members can interact with each other directly; there is no leader that all members must work through, and no leader explicitly orchestrating the team members' interactions. Instead, each team member must interact with each other as he or she sees fit, in keeping with that team members' understanding of the team's goals, plans, agreed upon methods of interactions, and assessment of other's needs and

---

[6] Graetz, Boyle, Kimble, Thompson, Garloch, 1998; Levine & Moreland, 1990; Hollenbeck et al, 1995.
[7] Noble, 2003a.

capabilities. In the taxonomy, this is a flat team having no designated leader, having a high degree of team member dependency, and making decisions as a group.

Though these kinds of teams are often highly motivated and successful,[8] they operate in a challenging environment, for without a leader able to explicitly direct activities, the team's success depends to a great extent on how well the individual team members know what they need to do. It is especially important in teams such as these that all members have the knowledge and understandings described below in Section 2.2.



Results Highest potential for constructive collaboration, but …success depends on all nodes interacting effectively for benefit of the team

"Control replaced by guidance and direction, giving highest degree of autonomy and initiative, but losing some predictability…. *

Figure 1. A Self-Managed Team with Peers

2.1.3 Team Development and Performance Measurement

Though there is a large diversity of types of teams, all share some common traits. For example, all progress through a common development cycle as team members work together, and all are responsible for some product or outcome that can be measured.

Most teams when they first form go through a period of adjusting as they get to know one another. As they progress, they usually improve and perform better. In the academic literature, this improvement is called team "hardening." The stages of hardening are called more colorfully in the popular literature "forming, storming, norming, and

[8] Kirkman, and Shapiro, 1997.

performing."[9] In our interpretation of team hardening, teams get better after first forming because they get to know each other better, and because they acquire a betting understanding of team goals and processes. That is, the knowledge that team members need in order to work together effectively; e.g., the knowledge described in Section 2.2, is getting put in place.

A second characteristic common to all teams is that their performance can be measured. In these assessments, we distinguish between bottom-line measurements and audit trail measurements (Figure 2). The bottom line measurements assess how well the team did in terms of their products and achievements. For an action teams, this bottom line measurement is often whether the team won. For thinking teams, it is the quality of such intellectual products as situation assessments, plans, recommendations, and decisions. The audit trail measurements assess team traits that can account for changes in team performance. The two main elements of the audit trail traits are team behaviors and team knowledge. This section reviews some product and behavior metrics. Section 2.2 will describe the knowledge important to measure in establishing a performance audit trail, and will describe some approaches for measuring this knowledge. The report "Command Center Performance Assessment System"[10] describes team metrics in considerable detail.

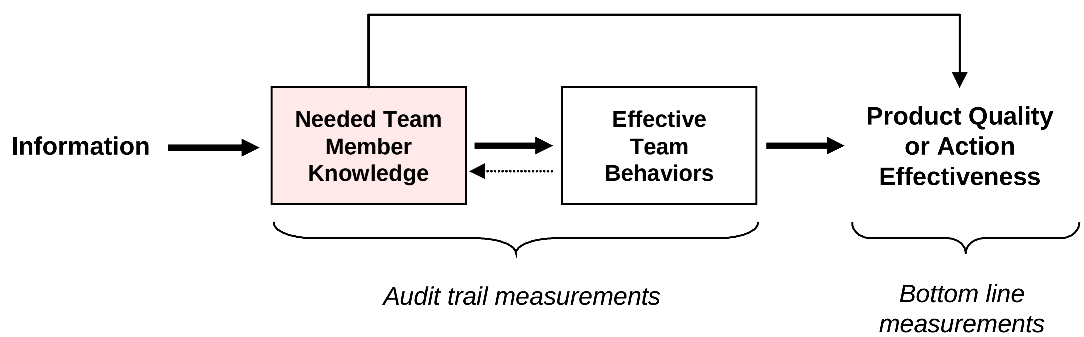


Figure 2. Bottom Line and Audit Trail Measurements of Team Performance

Three important products for collaborating teams are situations assessments, plans, and decisions. Table 2 summarizes metrics for each of these products.

Note that these product metrics are the same for teams as they would be for individuals. The measure of a product's quality just depends on the product itself, and not how it was created. Because it is practical to collect data needed to support these measures and because they address fundamental aspects of an intellectual product, these metrics have proved very useful for testing new technologies or procedures that are supposed to improve team performance.

- The situation assessment measures go beyond the surface level of knowing who's where in a situation, but also address the "deeper" understanding needed to decide

[9] Tuckman, 1965.
[10] Kirzl et al, 2003.

what to do next. Thus, these measures help explain why people make the decisions they do.

- The plan metrics check three key properties of plans. First, by measuring how long it is carried out, the plan durability metric measures plan robustness—its ability to tolerate changes in the situation. Second, checking the plan against its ability to support commander objectives measures whether the plan is suitable; e.g., whether the plan if successfully carried out will enable the team to achieve its objectives. Third, measuring contingencies checks whether the plan hedges for key uncertainties in the environment, an important strategy for handling uncertainty.

- The decision measures are somewhat tricky because they side-step the two most direct and conventional measures of decision quality: did the decision lead to a good outcome and did the decisionmaker use a "rational" deliberative process of identifying and evaluating several alternatives. Our measures de-emphasize outcome because outcomes depends too much on luck. Good decisions can lead to bad outcomes and bad decisions can lead to good ones. Our measures also avoid scores based on following a "correct and rational" decision process because experts often just jump to an excellent decision without following a formal rational process.

Table 2. Metrics for Situation Assessments, Plans, and Decisions

Product	Metrics
Situation Assessments	• **Correctness/completeness of estimates of location, identity, status, capabilities, and behaviors of forces** • **Compatibility of estimates for adversary intent and possible courses of action with adversary capabilities and behaviors** • **Correctness/completeness of identified opportunities and risks**
Plans	• **Useful life of a plan compared to its intended useful life. No plan "survives contact with the enemy," but better plans last longer** • **Fraction of commander's objectives that plan addresses** • **Fraction of plausible contingencies covered by plan**
Decisions	• **Extent that decisionmaker considers key factors: e.g., consideration of situation drivers such as centers of gravity, hedging for critical uncertainties** • **Expert rating of alternative selected**

Team behaviors often have a huge impact on team success. Measures of behaviors assess the extent to which a team is a well-oiled machine or whether the members are successfully emulating the Keystone Cops. Of course, we do not just attempt to quantify

an overall degree of keystone cop-ishness. Instead, we define specific behaviors that, like our measures of product quality, get to fundamental issues that really matter to team success. These specific behaviors can be grouped into three general areas: team members' activities, information handling, and the ability to adapt. Each of these general areas is associated with a set of more detailed behaviors (Table 3).

Table 3. Critical Team Behaviors

General Behavior Area	Behavior Elements
Team member activities	• **Right level of busyness** • **Effective coordination** • **Working on right tasks**
Information handling	• **Identifying needed information** • **Sharing with right people at right time** • **Effective leveraging of perspectives** • **Effective information organization**
Ability to adapt	• **Recognizing need for adaptation** • **Implementing the adaptation**

Actually quantifying how well a team is doing in each of these behavioral areas requires much more precise rating scales. The Evaluation Handbook[11] provides these scales. Table 4 shows what the scale for the behavioral element "identifying needed information" looks like. The measurement system divides this element into two parts, one dealing with the extent that a team understands what information is needed in order to solve its operational problem and another concerned with how well the team understands the operational perspectives it needs to consider. The "Specific Behaviors" column then tells what an evaluator should look for in order to evaluate how the team is doing with respect to a behavior element. For example, if "staff analyses and recommendations overlook important aspects of an operational problem," then the staff is not doing very well in identifying the information it needs. If it knew what information it needs, then presumably it would not overlook this problem aspect.

Table 4. Measurement Scales for Rating How Well a Team Is Able to Identify the Information It Needs

Dimensions of Behavior Element	Specific Behaviors For Measuring Behavioral Element
Operational Problem Understanding	**1. Staff analyses and recommendations overlook important aspects of an operational problem** **2. Staff members are unaware that a specific item of information was required to address a particular operational problem** **3. Staff analyses and recommendations exhibit an unnecessary**

[11] Kirzl et al, 2003.

	degree of uncertainty or reflect an unwarranted/erroneous assumption
Understanding of Stakeholder Interests and Perspectives	**1. Staff analyses and recommendations ignore a specific functional or organizational perspective** **2. Staff members are unaware that a specific area of expertise was relevant to a particular operational problem** **3. Staff members are unaware of specific stakeholder interests in a given operational problem area**

2.2 What People Need to Know and Understand To Work Together Effectively

This section discusses what team members need to know in order to work together effectively. It discusses some basic premises about knowledge and teamwork and then describes important functional categories of knowledge related to acquiring and sharing information and obtaining consensus. This section concludes by describing our twelve knowledge enablers. This discussion of knowledge enablers details the concrete knowledge and understandings that team members need, describing for each specific kinds of knowledge, what can happen when that knowledge is missing, how tell if it's missing, and ways to help teams acquire it.

2.2.1 The Importance of Knowledge

The starting point for describing the role of knowledge in effective collaboration is the following four basic premises. These outline the most important premises in this knowledge-centered theory of collaboration.

1. Knowledge is central to collaboration and teamwork. Teams whose members know what they need to know can work together effectively. Those that do not are prone to various kinds of predictable errors, with the type of error dependent on the type of knowledge deficiency. (Liang, 1995)

2. Knowledge must be distributed among members of a team. Everybody does not need to know everything for a team to be effective. But every team member does need to know how to get the knowledge he or she needs. (Wegner, 1987)

3. Individuals need to know about both "taskwork" and teamwork. Teamwork knowledge is what team members need to know to work together effectively . Taskwork knowledge is what team members need to know accomplish their part of the team's tasks. (Canon-Bowers, 1993)

4. The collaborative dialog helps generate the needed teamwork and taskwork knowledge. Team members exchange ideas to put in place the knowledge and understandings that team members must have for the team to achieve its mission. (Argote, 2000)

The first statement, that team members cannot work together effectively if they do not have the knowledge needed to do so, is our basic premise, and as written is almost a tautology. Section 2.2.3 makes precise what that knowledge is and what can happen when

it is missing. The second statement, that team members do not personally need to know all critical knowledge but do need to know who to ask to get the knowledge, is the basis of “transactive memory.” Sharing the responsibility for keeping track of various kinds of information is one of the biggest advantages of teamwork. The third item emphasizes that all teams are really working on two basically different kinds of issues: (1) creating their task’s products or performing task actions, and (2) maintaining team relationships. It is not enough for every team member to be an expert in their individual jobs for the team to succeed; team members also need to know how to work together. The last item addresses how team knowledge builds on itself. In teamwork, there is a kind of self reinforcing cycle. Knowledge is needed for teams to work together effectively, but teams need to work together in order to obtain this knowledge.

Figure 3 illustrates the relationship between knowledge (the “individual and shared understandings”) and some key team activities: “team set up and adjustment,” “group problem solving,” and “synchronize and act.” This diagram helps show how knowledge success begets success and small failures grow into big ones.

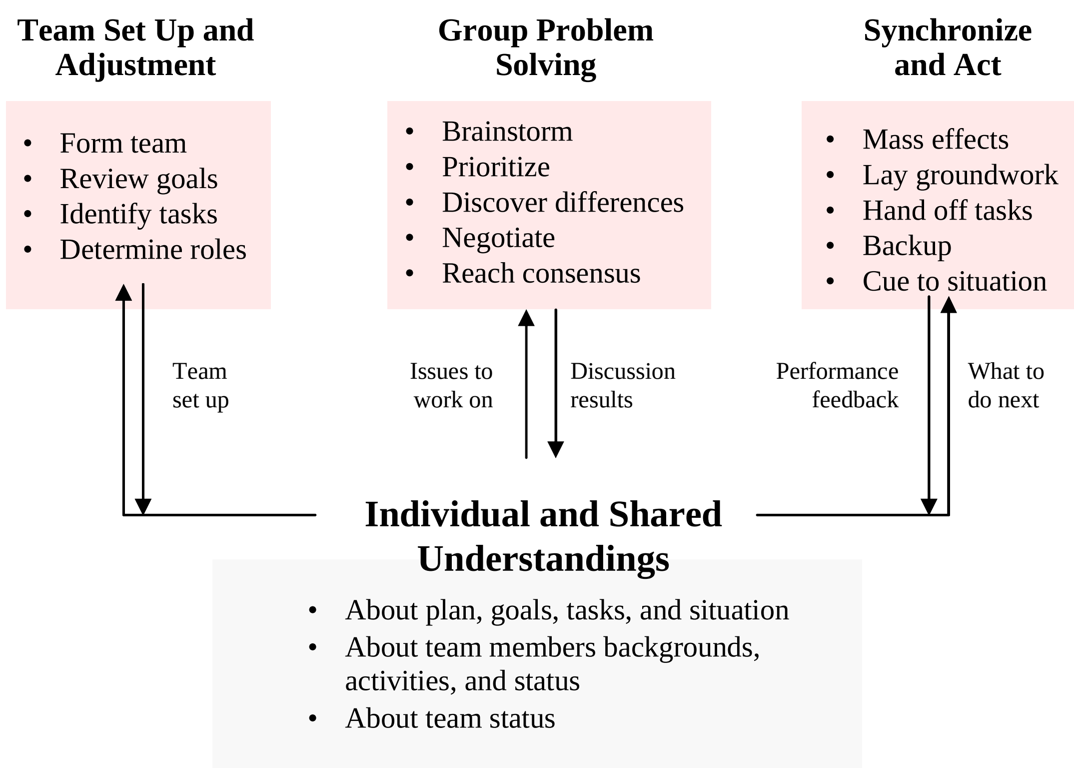


Figure 3. Relationship between Knowledge and Team Activities

All teams perform all three of these activities, generally moving from left to right but also switching back and forth among activities according to their immediate needs. In “set up and adjustment” the team organizes itself, reviewing goals, allocating roles and tasks, and defining the team’s business rules. In the process of doing this, they generate and deposit critical team knowledge about goals, tasks, roles, and team interaction methods. Some of this knowledge may be written down in team documents, but much of it will reside as

tacit knowledge in team members' minds. Team members need this knowledge when they carry out their "group problem solving." Here they identify and critique different issues, discover differences and align understandings, negotiate, and reach consensus about the nature of the problem and what they should do. When doing this, they draw on the knowledge acquired while performing earlier team tasks. As they progress, they refine and augment their knowledge with the results of their work. The same sequence also occurs with "synchronize and act." Team members draw on their knowledge to coordinate and help each other. They deposit knowledge about what works well and how they should interact as they work together.

This work-knowledge relationship can create highly damaging action-knowledge cycles. A small amount of missing knowledge can undermine a team activity that creates information critical to later team functions, and when missing, causes these later functions to fail. Thus, it is important for teams to catch these small knowledge gaps quickly, before they grow and cause significant damage. The theories in this report lay the groundwork for enabling teams to do this. The Collaboration Advizor™ Tool, described in Section 3, organizes and presents needed team knowledge to help teams diagnose their knowledge problems and decide how to fix them.

2.2.2 A Knowledge Framework

This is the first of two sections discussing the kinds of knowledge that team members need in order to work together effectively. This section talks about some theoretical classes of knowledge that people need in order to integrate new information into their current understandings, to decide when they should get additional knowledge or share information with others, and to decide when they need to encourage additional discussions to obtain consensus about a team's views, positions, and decisions. The next section describes the specific concrete knowledge content that team members need.

Though we sometimes talk about "team knowledge" or "team understanding," in our theory, these terms are just metaphors. In fact, all of the team understandings reside in the minds of the individual team members. This section describes the cognitive structures that can make it appear as if the collection of knowledge in individual's minds functions as if it's "team knowledge." It addresses the knowledge structure we hypothesize that people use when they (1) integrate new information into current understandings; (2) estimate what others know and where the team stands; and (3) decide when to share information and work on consensus.

Integrating new information into current understandings

When people "understand" new information, they integrate this new material into existing knowledge structures.[12] These knowledge structures, called "schema" in the reference, organize knowledge into beliefs about how the world works and how actions can impact the world. Thus, new information does not write on a blank slate. Instead, it edits a web of preexisting beliefs. What people initially believe has a big impact on what

[12] Noble et al, 1989.

they believe after they receive new information. Successful transfer of meaning, sending information to update a recipient's understandings, requires that this editing process produces a result consistent with the meaning that the sender of the information intended to convey. This report calls the cognitive processes in which existing understandings are used to interpret new information the "default reasoning model" (Figure 4). Key features of the default reasoning model are:

1. When understanding information, people continually form hypotheses about what is happening. In the default reasoning model, each of these hypotheses is represented by schema. Each schema contains a set of slots that specify current beliefs about various characteristics of the situation. These schema can be formed several different ways, including composition from component schema, specialization from more abstract ones, and abstraction/induction from more specialized ones.

2. Before any information is available, people's schema contain default values that represent their beliefs about the characteristics of different types of situations. These are inherited from the long-term knowledge. Usually, people differ in these default values. They generally differ more when they have different backgrounds and usually differ the most when they come from different cultures. People's ability to predict other's default values improves as they get to know one another.

3. When people receive information, they combine this information with their current beliefs, thereby updating their current beliefs. Thus, people's current understandings reflect both the newly acquired information and their previous beliefs, which schema default values strongly impact. Differences in default values explain much of the differences in people's interpretations of the same information.

4. People use various heuristics and cognitive workload saving strategies when they update their current beliefs. Side effects of these heuristics and strategies can manifest themselves as cognitive biases. For example, in the confirmation bias people look only for information able to confirm their current hypotheses. This heuristic saves work by preventing the creation of new hypotheses. It has a bad side effect however when it prevents the creation of a new hypothesis that happens to be correct.

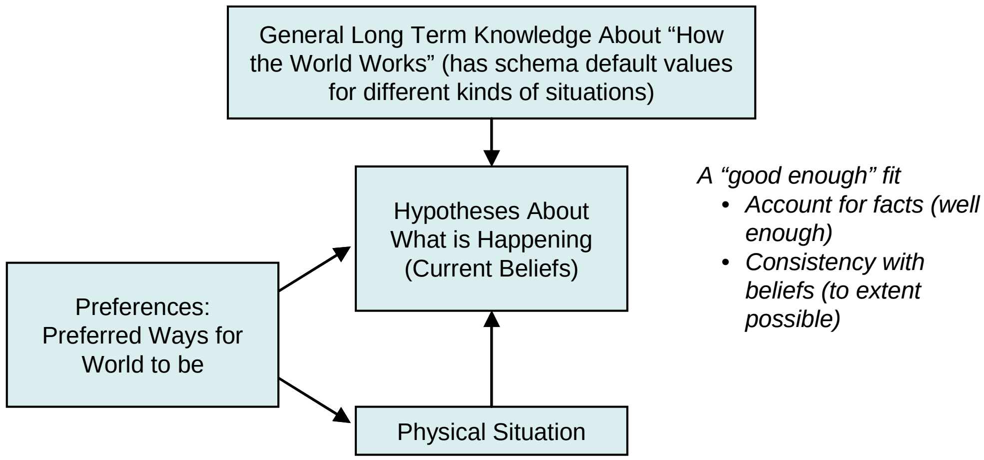


Figure 4. Default Reasoning Model

This notion that people understand new information in terms of what they already believe impacts many facets of successful teamwork. It explains the importance of knowing other team members in order to communicate effectively. It explains why people in teams whose members come from different cultures or have different backgrounds have more difficulty understanding each other. It establishes a basis for understanding some of the significant ways that people can misunderstand information. And most significantly, it explains the basis of the team's cognitive glue—that knowledge that enables each team member to accurately estimate what their teammates know and believe, and so helps that team member to know what information and beliefs need to be shared, discussed, or worked on.

Understanding what others know and where the team stands

Figure 5 organizes knowledge that helps glue together the individual team members. As emphasized earlier, all the team knowledge in a "team mind" actually resides in the minds of the individual members. Thus, because the specific content of the knowledge in Figure 5 varies among team members, there would be a separate diagram for each team member.

This model organizes knowledge along three dimensions. The horizontal axis is the declarative and procedural knowledge that team members need in order to work together effectively. Section 2.2.3 describes this knowledge in detail. The vertical axis groups knowledge in ways related to consensus and alignment.

The third dimension, the axis into the page, draws on the Default Reasoning Model just discussed. It addresses team members' current beliefs and the reasons for team these current beliefs, the default beliefs, and the evidence supporting current beliefs. Note that in the absence of evidence, team members will assume their defaults while hedging for other possible interpretations.

The model identifies three kinds of knowledge along the horizontal axis. It makes the usual distinction between declarative knowledge (e.g., facts) and procedural knowledge

(how to do things). Within declarative knowledge, it distinguishes between estimates of what is vs. the “image” of what the team member would like the status to be.[13] Comparing the status of the team and tasks with beliefs about what the status should be is extremely important to any decisions about changes that need to be made.

The vertical axis is perhaps the most important for understanding the relationship between knowledge and collaboration processes. These processes include ways for updating one’s own understandings of tasks and teams, for sharing information, for obtaining a common view about the team and task, and for reaching a consensus about team goals, the situation, and desired team actions.

The first row, “own individual knowledge/beliefs,” is what every team member believes to be true about the task, external situation, plan, goals, etc. These are the understandings and current beliefs that result when people update prior beliefs with new information. The second row, “estimates of others’ knowledge,” is a person’s estimates of other people’s understandings and current beliefs. It includes estimates of what people know, what people would like, and what people know how to do. This is the most important row for sharing information, for people provide information when they believe the recipient needs the information and does not already have it, and they request information from the people they think have it or know how to get it. The third row specifies that team member’s estimates about the alignment of understanding. This is the knowledge of the extent to which team members agree or disagree, including also understanding of the reasons for disagreements. The fourth row is the team member’s views on the status of team consensus. These views include his understanding of what the consensus is, the extent to which team members buy into this position (how much of a consensus there actually is), and his opinion of whether the consensus position is a good one.

[13] As in image theory. See: Beach and Mitchell, 1987.

Default beliefs

Evidence

Current beliefs

	Declarative knowledge	Declarative evaluation knowledge	Procedural knowledge
Own individual knowledge/ beliefs	Taskwork and teamwork knowledge as specified by twelve enablers	Evaluation of adequacy of that knowledge	Means to improve knowledge
Estimates of other's knowledge	Estimates of what others know, including their evaluation of their knowledge	Evaluation of what others know	Means to confirm what others know. Means to improve what others know
Estimates of alignment of understandings	Areas of agreement and disagreement, and reasons for differences	Importance of increasing alignment and resolving disagreements	Means to increase alignment, including ways to identify reasons for differences
Perceptions about consensus	What team has agreed to and extent of buy-in	Adequacy of consensus position of obtaining team goals. Adequacy of degree of buy in for team coherence	Means to improve consensus; e.g., negotiation

Figure 5. Knowledge Categories Important for Team Decision Making

Deciding when to obtain or share information and work on consensus

ONR and NAVAIR have developed another model that focuses on processes central to teamwork and collaboration (Figure 6). These central processes are knowledge construction, collaborative team problem solving, team consensus, and outcome evaluation and revision. In accomplishing these processes, team members obtain, share and discuss information, and work toward achieving consensus. The knowledge framework described above helps explain how people decide when to engage in each of these processes.

STRUCTURAL MODEL OF TEAM COLLABORATION (summarized)

COLLABORATION AND KNOWLEDGE MANAGEMENT (CKM) PROGRAM

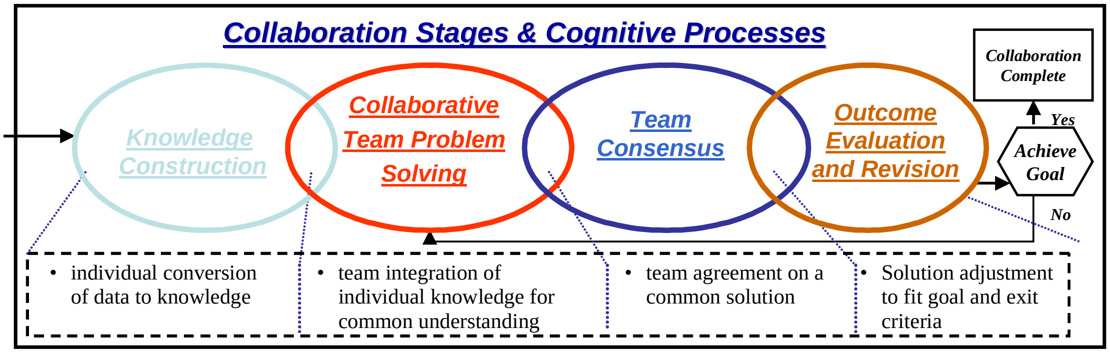


Figure 6. Structural Model for Critical Collaboration Processes

Team members decide to engage in each of these processes either when their plan or standard procedures call for it, or when they decide it is necessary even though not explicitly called for by either their plan or standard processes. The first criteria for choosing to do something—that it is called for by the plan or is the normal way the team does things—is straightforward. The second reason, that it is now necessary, is more complex. This decision to do something because it is necessary to do it even though not explicitly prearranged is called "opportunistic decisionmaking."

Though applicable to cognitive modeling, this model was first popularized in the early DARPA Hearsay program that developed technology for understanding spoken English. The Artificial Intelligence community has employed this model extensively using a *blackboard* architecture.

This model assumes a set of knowledge sources (e.g., human or computer team members), each having special skills for contributing to the development of the collaborative product. All knowledge sources deposit the results of their work on a blackboard that all can see. When a knowledge source sees information posted that he/she/it can process further, the knowledge source takes that posting as input, carries out its specialized processes, and deposits the answer back on the backboard, where it can provide input to other knowledge sources (team members).

The process is called *opportunistic control* because there is no set sequence in which the team members contribute to product development. Instead, each team member contributes when the opportunity arises; e.g. when some other team member deposits information on the blackboard that that team member can use. Because deposited information might be of immediate use to several team members, blackboard architectures also include rules for breaking ties—for deciding which team member gets the chalk. In the Knowledge Enabler model, multiple team members can work in parallel, so breaking ties is not as important. Ties that need to be broken are decided using the team's plan, its business rules, and by team members and leader(s) understanding of the plan dependencies.

In real teams, the "blackboard" is a combination of explicit information and tacit knowledge. The explicit information resides in the documents, database, and visualizations available to team members. The tacit knowledge resides in the heads of the team members.

To make opportunistic control work, team members need to know when a situation has arisen that calls for some action not explicitly specified by the plan. The knowledge category in Figure 5 most important for determining this is the declarative evaluative knowledge. This knowledge is each person's understanding of the adequacy of some knowledge important for teamwork. For example, a person's deciding that his individual knowledge is inadequate can trigger a search for additional information. A person's deciding that some other team member's knowledge is inadequate can trigger a decision to provide that person with some helpful information. A person who decides that his understanding is insufficiently aligned with another team member's may initiate discussions to increase that alignment. A person who decides that the team's consensus position is weak may voice his concerns about that position.

This abstract framework, though very useful for understanding basic kinds of knowledge needed for team decisionmaking, does not specify the specific concrete knowledge that teams actually need in order to work together effectively. There is a vast amount and diversity of concrete knowledge—all of which can be critical to team effectiveness under some conditions. The following section discusses this specific information, organizing it into "twelve enablers," describing when it matters, what might happen when it is missing, how to tell when it is missing, and identifying ways to fix knowledge problems.

2.2.3 The Twelve Knowledge Enablers

The collaboration enablers organize the knowledge that teams need in order to work together effectively as a team and produce a good product. There are many different ways to organize needed team knowledge.[14] We chose the twelve to be described below because they (1) provide a level of diagnosis that points to concrete actions able to improve team performance; (2) are easy to understand; (3) map reasonably cleanly onto risks for and symptoms of team problems, and (4) as a set, account for key team behaviors, such as team agility, team member backup, accountability, and coordination. The Collaboration Advizor™ Tool, the expert system that helps teams diagnose and fix knowledge-based collaboration problems, builds on this organization of knowledge. This organization has worked well for this tool. All five of the teams that tested the tool at the time of this report have found this knowledge organization useful.

Figures 7 and 8 provide a snapshot summary of the twelve enablers. For each enabler, the Figures describe the type of knowledge that that enabler addresses. It then lists the "sub-enablers" for that enabler.

[14] Matheiu et al, 2000.

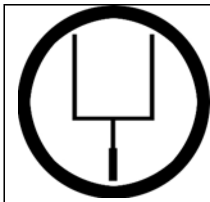
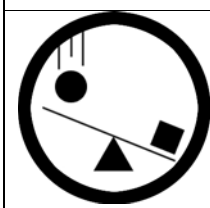
	1. Goal understanding. Knowing what the customer wants. a. *What client wants* b. *Criteria for evaluating team product*
	2. **Understanding of roles, tasks, and schedule**. Knowing who's supposed to do what and when, and with what information and resources. a. *Tasks* b. *Roles and role backups* c. *Schedule* d. *Assigned resources/specified information needs* e. *Assumptions and contingencies* f. *Criteria for evaluating task progress*
	3. **Understanding of relationships and dependencies**. Knowing how entities, events, and tasks impact the plan. a. *Task to task* b. *Task to goal* c. *Situation to plan* d. *Resource/information/labor to tasks*
	4. **Understanding others.** Knowing what other team members' backgrounds, capabilities, and preferences are. a. *Others capabilities/knowledge* b. *Others values/preferences* c. *Others work habits* d. *Others motives/rationale for working on project*
	5. **Understanding of team "business rules."** Having and knowing effective and agreed upon rules for team members interacting wit each other. a. *Team etiquette* b. *Team policies, like how to resolve conflicts*
	6. **Task skills**. Knowing how to do one's assigned work. a. *Mechanics of performing tasks* b. *Using support tools* c. *Finding needed information* d. *Getting help*

Figure 7. First Six Knowledge Enablers: Team Preparation

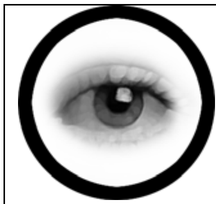
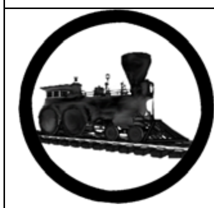
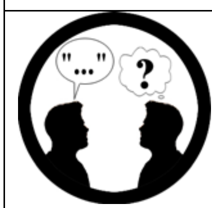
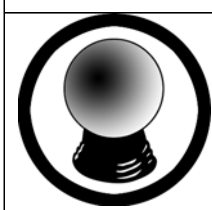
	7. **Activity awareness**. Knowing what others are doing now and current need for doing it. a. *What others are doing* b. *Their level of engagement and busyness* c. *Whether it is work that the team member should be doing (including for self)*
	8. **Understanding of the external situation**. Knowing status of people (including client), things, and events of the world outside of the team and projecting future changes. a. *What's happening* b. *What's significant to team and why* c. *Projecting future events*
	9. **Current task assessment**. Keeping tasks on track, knowing how well own and other's tasks are progressing, and when to offer help. a. *Task status: who's on it; work, resources and information status* b. *Signs of problems, and if task is on track* c. *Task needs: advice, information, resources, labor*
	10. **Mutual understanding**. Knowing what other team members understand now and knowing if they agree or disagree. a. *Transfer of meaning--how others understand input* b. *Where others agree or disagree and why*
	11. **Plan assessment**. Predicting whether the plan will still enable the team to achieve its goals. Knowing a. *Inter-task projections: resources, people, information availability* b. *Plan prospects and danger points*
	12. **Understanding of decision drivers**. Judging and applying the criteria for selecting an action. Knowing: a. *the right factors to consider* b. *deadline for decision needs to be made* c. *how to manage uncertainty* d. *right people to involve*

Figure 8. Last Six Knowledge Enablers: Status Assessment and Decisionmaking

These twelve enablers fall into two major groups, displayed separately in Figures 7 and 8. The first six, "team preparation," comprise the foundational knowledge that tends to build and change slowly over time. This is the knowledge that accounts for team members being able to work together more effectively as they get more experience working together. The second six, status assessment and decisionmaking, are the "real time" knowledge and understandings that can change dramatically instant to instant. This is the

knowledge and understandings that enable people to react quickly to changing circumstances.

The twelve enablers extend previous models of individual performance and command and control to teams. Several of the enablers—task skills, plan assessment, and the understandings of goals, the plan, dependencies, the external situation, and decision factors—are as essential to individual work as to teamwork. The detailed knowledge for these enablers differs very little from that required for individual performance. Several others of the enablers apply only to teams and are not relevant to an individual working alone. These are the enablers for understanding others, team business rules, activity awareness, and mutual understanding. One enabler, "current task assessment" is not only important when working alone, but also imposes significant additional knowledge requirements in a team setting.

The twelve enablers are not independent, but build on each other. Figure 9 depicts a few of the most important dependencies. As an example, "mutual understanding" (knowing the extent to which team members agree or disagree) depends heavily on "knowing each other" (which includes knowing a person's background, capabilities, and general preferences). Knowing each other, in turn, depends on task assessment (knowing the status and progress of tasks) because one way to get to know another team member is to see how they perform on tasks.

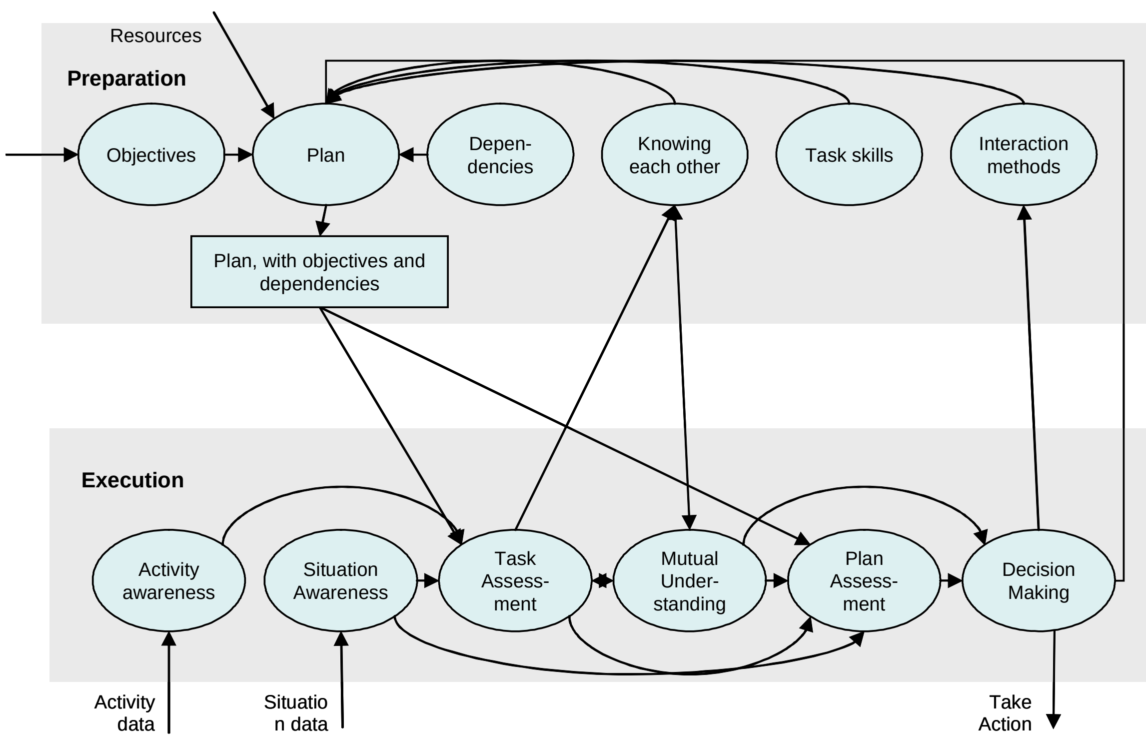


Figure 9. Enabler Dependencies

The remainder of this section elaborates on these twelve enablers. For each enabler, it (1) discusses some general aspects of the enabler, including the specific knowledge it

includes; (2) describes the importance of having that knowledge, the consequences of not having it, and the circumstances when having it is especially important; (3) outlines ways to acquire this knowledge and the impediments that hinder obtaining it; and (4) describes how to determine, either by asking questions or by observing behaviors, whether a team needs to improve its knowledge for that enabler.

Enabler 1: Goal Understanding

Knowledge. Goal understanding is the first, and perhaps the most important of the knowledge enablers, for teams that do not know where they are going are unlikely to go there. It includes:

a. *What the client (commander in military) wants*
   i. Client goals and "real" requirements
   ii. Client stated requirements
   iii. Client's current concerns
   iv. Client culture
   v. How to interpret feedback from client

b. *Criteria for evaluating team product*
   i. Criteria for being done
   ii. Criteria for product quality
   iii. Criteria for mission success

Goal understanding includes understanding both the explicit and implied goals of the team project, taking into account the values and cultural norms of the tasking authority. It should be thorough enough so that individual team members can modify planned actions appropriately in anomalous unexpected situations.

Team members who understand goals know the reasons that the team was tasked, the products or actions desired, and the criteria for good products and effective actions. They know the specific measures for evaluating team and individual success. They also know the client's constraints on how the team may behave and on the permitted characteristics of the product. These include both explicit constraints such as military rules of engagement and implicit ones associated with the culture in which the team operates.

Importance of goal understanding. Understanding goals, objectives, and measurable indicators of success is critical for organizing the team, developing a plan, setting priorities, establishing individual and team accountability, and assessing progress. People who understand goals can estimate the desirability of various outcomes and side effects of actions. They can also understand the tasks implied by these goals.

A team that does not understand its goals is at risk for (1) creating a product that does not meet customer needs or (2) taking actions that do not support their mission. When individuals do not understand goals, they cannot prioritize their individual tasks appropriately nor properly shape these tasks as circumstances change. Consequently, they may not take the actions required to fully support the team.

Understanding goals is always important for every team. It is likely to matter the most when anomalous unanticipated situations arise and team members cannot receive timely feedback needed for mid-course corrections, either because there is no time for feedback, or because the customer is not available or does not want to be bothered.

Obtaining goal understanding. The most direct way to understand explicit team goals are briefings and documents stating these goals; e.g., written plans and requirements traceability documents. Interactions with leaders (e.g., military commanders) and clients help convey both explicit and implicit goals, especially when non-verbal cues may be communicated. For example, a client critique of proposed actions, as happens during planning, helps team members understand goals. This is the reason that participating in planning helps people understand goals. Knowing the leaders, clients, and their cultures helps people understand implicit goals. Group discussions of specific success criteria, especially in terms of the properties of desired team products, contribute to goal understanding.

Factors that can impede goal understanding include unclear communication of goals from customers and team managers, remote customers or team members who are difficult to reach, new customers, team members who are unfamiliar with a customer's business area or culture, and presence of multiple stakeholders or of multiple competing goals. Even if team members understand the goals themselves, they will have difficulty applying this understanding in deciding what to do if criteria for determining mission success, product quality, or task progress are unclear.

Evaluating goal understanding. Observers can estimate how well team members understand goals by asking them questions and by observing their behaviors. They can ask team members what team goals, constraints, and success criteria are. They can probe for a deeper understanding of client priorities by asking team members to evaluate the desirability of specific hypothetical outcomes under various circumstances.

Behavioral indicators of poor goal understanding are the return of products for revision, arguments or disagreements about what achievements constitute success, and actions (or failures to take action) which the client believes are inconsistent with his or her intent. Other behaviors indicative of poor goal understanding are team members pursuing their own objectives rather than supporting team needs, or team members proposing actions which if successful would be inconsistent with intent.

Enabler 2: Understanding of roles, tasks, and schedule

Knowledge. This is the "surface" understanding of the plan that describes how the team is going to achieve its goals. Teams without a plan can be very chaotic, for nobody knows

what they are supposed to do. Plans can be very detailed, as is often the case in the military. Alternatively, they can be informal, not even written down but understood by all team members anyway.

Project plans usually decompose the team's work into separate tasks, assign these tasks to separate individuals or groups of people, and then specify a schedule. The plans often identify team member responsibilities, to include both fixed and context dependent leadership roles, principal task performers, and task backups. A highly detailed plan will explicitly address all of the following. However, even if there's no detailed listing of all of these elements, people in effective teams have this knowledge.

- *a.* *Tasks*
    - i. Task purpose
    - ii. Task details—what it entails
    - iii. Exceptions—when to modify
- *b.* *Roles and role backups*
    - i. People assigned to tasks
    - ii. People responsible for different areas of knowledge
    - iii. How each person contributes to the team
    - iv. Backup responsibilities
- *c.* *Schedule*
    - i. Task deadlines
    - ii. Intermediate milestones
- *d.* *Assigned resources/specified information needs*
    - i. Resources assigned to tasks
    - ii. Information required to guide doing a task
    - iii. Information required for integration into a product
- *e.* *Assumptions and contingencies*
    - i. Assumptions that must hold for plan to work
    - ii. Contingent actions to take when assumptions do not hold
- *f.* *Criteria for evaluating task progress*
    - i. Criteria for moving to next task
    - ii. Criteria for successfully completing a task

iii. Criteria for determining progress of task

Note that team members need to know not only their own tasks, but also related tasks that are assigned to other team members. Thus, plan knowledge includes understanding the tasks that each team member is primarily responsible for, tasks that sub-teams are jointly responsible for, team member back-up responsibilities, and tasks that they depend on or that depend on them. Plans also include "coordination conventions" (like stop signs). These are special signals, well-defined language, or markers that help team members coordinate their work.

Importance of understanding the plan. Team members need to understand their plan in order to know what they are responsible for and to know whom to talk to when they need to discuss tasks that they depend on or support. They need to understand the schedule so they know when they should perform their tasks. Assumptions and contingencies are about having a "Plan B" to be used in case their primary plan is not working. The criteria for plan progress enables team members to know whether they are on track, need to make minor adjustments, or switch to their Plan B. Other enablers, particularly understanding of dependencies, supports the "deeper" understanding of the plan needed for projection, prediction, and adjustment.

Team members who do not understand individual roles and responsibilities may incorrectly work on tasks that are assigned to someone else, may fail to work on a task that they should be doing, may fail to coordinate with team members they should be working with, or may fail to provide the help or backup that another team member needs. Team members who do not know the schedule may perform the tasks too late to be useful.

Understanding of roles, tasks, and schedule matter the most when tasks are highly interdependent so that poor performance of one task seriously undermines performance on others. It is also especially important when the plan has little slack and tasks must be done immediately with little time for task discussion, when team members cannot easily contact one another to discuss roles, or when a team member becomes unable to carry out his assigned tasks. Understanding criteria for task progress is especially important when anomalous unanticipated situations arise that require new tasks.

Obtaining understanding of roles, tasks, and schedule. Perhaps the best way to understand the plan is to participate in its creation. This helps not only in understanding the plan itself, but also in understanding the reasons for the plan. If the plan stops working, this understanding speeds up the needed re-planning. Rehearsing the plan and discussing how to handle various problems that might arise helps people understand plan contingencies. Plan documents, especially when they include such diagrams as Gantt or Pert charts help teams know the schedule. Knowing team member capabilities and role requirements helps people infer roles, as does being aware of what people do as the team progresses. Experience working together helps team members know team roles and tasks.

Several factors make it harder for people to understand the plan. If the plan is complicated, having many tasks, tasks that could be assigned to several different team

members and tasks that are the joint responsibility of several team members, then it is harder to understand the plan. This is especially true if the plan was never written down, if a work breakdown structure was never developed or if no graphs or diagrams depicting the schedule are available. Additional factors are failure to clearly define member roles and responsibilities or failure to assign backup responsibilities. It is harder to understand roles in teams with more than ten to twelve members. Because roles can be inferred by watching people work, new teams where people have not had a chance yet to see each other work or infrastructures in which it is hard to watch people carry out their roles impede others from understanding these roles. Finally poor communications that hinder discussing roles, task, or schedules impede this understanding.

Evaluating plan understanding. The most direct way to find out how well people understand the plan is to ask them to describe tasks and team member roles, including backup responsibilities. One can also ask what the schedule is, including task start and completion times, and ask about specific criteria to be used in determining assignments to new unanticipated tasks.

Behavioral indicators of poor task and role understanding are disagreements on the team about who is responsible for various kinds of work, tasks that no one performs even though needed resources are available, tasks that two people do redundantly contrary to the plan, and team members when asked to help saying that they are not the right person to ask. Indicators of not knowing the schedule are tasks started or completed too late to be useful. Failure to understand planned coordination conventions are indicated when team members do not respond to a team coordinating signal or respond in ways that are no longer appropriate.

Enabler 3: Understanding of dependencies between tasks, resources, time, information, and the situation

Knowledge. This is the "deeper" understanding required to project success and make adjustments. It includes the logical, temporal, resource, and information relationships among tasks. This knowledge is sometimes called the situation and plan "mental models"—models needed to reason about the plan and situation, to project futures, and to make decisions.

This enabler includes knowledge about dependencies among goals, tasks, information and resources, and the situation. In the case of task dependencies, this enabler specifies how one task impacts other tasks and team goals and how various properties of the situation impact task viability. In the case of resources and information, it specifies the information and resources required to carry out a task and the consequences should these resources or information not be fully available. In the case of the situation, it links situation causes to effects, enabling decisionmakers to assess the consequences of possible actions.

Enabler 3 includes the following elements:

a. *Task to task*
   i. Task products that other tasks need

   ii. Impact of poor performance on other tasks

   iii. Impact of late completion on other tasks

b. *Task to goal*
   i. Goals tasks support

   ii. Impact on goals from different levels of task performance

c. *Situation to plan*
   i. Impact of environment changes (e.g., weather, neutrals, public)

   ii. Impact of adversary/competitor

d. *Resource/information/labor to tasks*
   i. Impact of resource shortage

   ii. Impact of information lack

   iii. Impact of skill or knowledge lack

   iv. Impact of labor unavailability

Importance of understanding task dependencies. The knowledge in this enabler makes it possible to predict effects from causes. It also enables people to predict what future tasks are at risk or may be delayed if a task is omitted, performed poorly, or delayed, or if needed resources and information are not available. By linking tasks to the situation properties needed for a plan to work, this enabler helps people understand plan contingencies.

Knowledge of task dependencies is key to carrying out a plan. Knowledge of logical dependencies, coupled with an understanding of goals, enable team members to prioritize work on competing tasks. Knowledge of temporally dependent tasks is key to synchronization. Knowledge of information dependencies drives information requirements, enabling people to identify needed information as the plan unfolds and the situation changes. Knowledge of resource dependencies drives logistics planning and supports decisions about task trade-offs or task reshaping. Knowledge of situation and task dependencies enables decisionmakers to evaluate alternative actions and determine whether a current plan will still enable the team to achieve its objectives. Team members who do not understand dependencies cannot predict the consequences of the situation, resources, or information on task performance, and cannot foresee the consequences to future tasks and goals on current task performance. Lacking such understandings, people may have difficulty prioritizing work, identifying information and resource requirements, or adjusting their work to changes in the situation.

Of course, understanding task dependencies matters only if the plan has these dependencies. When it does, then these misunderstandings are most significant when recovery from misunderstandings is the difficult. This occurs when the plan has little slack, when tasks have tight deadlines, when the team is spread over multiple locations, when team transparency is low, and when resources and information requirements are situation-dependent and not easily satisfied. Misunderstanding the situation dependencies especially matters if the team has an active intelligent opponent; e.g., there is an adversary in military campaigns or competitors in business.

Obtaining an understanding of task dependencies. A written plan, especially when accompanied by a synchronization matrix or other charts showing the logical and temporal task dependencies (such as a Pert chart) clarifies plan dependencies. Resource plans (information collection or logistics) help people understand those dependencies. In addition to having these documents, it is especially helpful if the team members who carry out the tasks also helped plan them. It also helps if the team members rehearse or work through the plan execution, reviewing the task flow and seeing the dependencies under various circumstances. Knowledge of situation dependencies usually requires people to be familiar with the situation, a familiarity normally achieved through experience and training. Influence diagrams or other graphical representations that link causes to effects help people understand these dependencies and so help them predict effects from causes. In unfamiliar situations, this knowledge can be obtained by systematically exploring the environment, taking actions in order to observe their results.

There are many factors that make it harder for people to understand these dependencies. These include complicated plans with tightly coupled dependent tasks, tasks that are extremely sensitive to the success of other tasks, tasks whose information requirements are sensitive to situation details and tasks that must compete for limited resources. Incompletely specified plans also impede this understanding. These include plans that do not specify contingencies for significant possible events, or do not clearly specify the information or resource requirements under various circumstances. Understanding of dependencies is also hindered when people responsible for carrying out the plan did not participate in the planning process or when team members never mentally rehearsed the plan.

Evaluating understanding of dependencies. There are many good questions to ask to determine how well team members understand dependencies. The most direct ones ask team members to describe task dependencies, task resource and information requirements, "long poles in the tent" that require advance preparation, and other factors critical to plan success. Less direct ones can ask about workarounds or substitutes for resources that may not be available. They may also ask about the information that other team members need to carry out their work, and about the consequences of specific task delays, omissions, or performance shortfalls.

There are many behavioral indicators of poor dependency understanding, but unfortunately most of these are ambiguous and can have additional cognitive or social causes. One behavioral indicator of poor dependency understanding is team member mistakes in prioritizing activities. A team member may choose to work on tasks that are

not urgent, ignoring others critical to other team members, or may cause another team member to do poorly by failing to finish a task in time. Additional indicators are team members that consume or capture a resource not essential to them but critical to someone else, that make requests that require unrealistic response times, or that make decisions or approve actions too late to be useful. Still other indicators are team members that fail to provide help (resources, information, labor) proactively to other team members, causing those team members to ask for such help, or team members that fail to react to situation cues with significant impact on their plan. Poor valuation of information importance can also indicate poor understanding of dependencies. Team members may fail to ask for essential information, or may focus on relatively unimportant information.

Enabler 4: Understanding of team members' backgrounds and capabilities

Knowledge. Team members need to know each other in order to work together effectively. Team members need to know each other's capabilities and knowledge, their values and preferences, their work habits, and what they hope to achieve personally by being on the team. More specifically, they need to know:

a. *Other's capabilities/knowledge*

- i. Other's level of expertise in various areas
- ii. Other's procedural knowledge in various areas; e.g., work skills
- iii. How to explain things to another so he/she will understand
- iv. Other's specific experiences and reactions to these experiences

b. *Others' values/preferences*

- i. Criteria for taking an action
- ii. Likes and dislikes
- iii. Ideologies
- iv. Preferred approaches and solution methods

c. *Others work habits*

- i. Thoroughness/attention to detail/care
- ii. Perseverance
- iii. Manner of dealing with others
- iv. Reactions to setbacks

d. *Others motives/rationale for working on project*

- i. Personal goals

For this enabler, the “team” can include people within a sponsoring organization that can contribute in some way to the team, such as by providing resources.

Importance of understanding other team members. Team members need to know one another in order to interact with each other efficiently. Knowing another team member’s background and experience helps people to know what information that team member needs, helps them know how to present that information in ways that the other team member can best understand, and helps them to know how to frame issues to improve cooperation. This knowledge is also needed to predict what others will do under various circumstances, to predict when team members may need help, and to know how to provide that help. It is also essential for establishing trust because people need to know what others can do or are willing to do under various circumstances. Knowing one another helps prevent surprises, where one person takes an action that is unexpected by others.

Team members who do not know each other well are much more likely to have undiscovered differences in viewpoints. When team members are not familiar with each other, they may seek help or information from the wrong people, fail to provide help and information to team members who need it, or count on someone to take an action that he or she does not know how to do or are unwilling to do. Not knowing one another can lead to confusion or disagreements. One may fail to explain issues in ways that will be understood, may fail to discover differences in understanding, or may fail to reach agreement or consensus.

Understanding others is almost always important. It is especially important when the team needs to improvise and modify the plan in order to respond to changing circumstances. Physically dispersed teams, especially those where it is difficult to know what others are doing, increase the chance that team members will take unpredicted and counterproductive actions discovered too late to correct. Teams that require task improvisation or joint activity to succeed also increase the adverse consequences from team members not knowing one another.

Obtaining an understanding of other team members. The best way for team members to understand one another is to have worked together in the past. People get to know one another as they work together, particularly if they can share views, see behaviors, and evaluate outcomes. People can also learn about each other’s backgrounds and experiences. People who share backgrounds and cultures can make better assumptions about each other than can people from different backgrounds and cultures.

The most significant factors that make it more difficult for team members to know one another are large teams (more than ten to twelve members), new teams, teams with changing membership, teams whose members have not worked together before, and teams whose members have different cultural and work backgrounds. It is harder for people to get to know one another on teams where it is difficult to communicate or difficult for people to watch each other work, share views, communicate non-verbal cues, or link team products or action outcomes to the people who did them.

Evaluating understanding of other team members. Evaluators can ask about any items in the list at the beginning of the discussion on this enabler. One can ask about the backgrounds of individual team members and about each person's areas of specialization. One can also ask who should be approached for certain kinds of information and assistance.

Significant behavioral indicators are team members asking the wrong person for help or information, team members needing to explain issues several times, and team members who take surprising actions in response to a request. Team member failure to persuade another to take an action is sometimes an indicator, for this can imply that the persuader does not understand the other's decision criteria.

Enabler 5: Understanding of methods for team member interactions

Knowledge. This enabler is knowledge about the team's "business rules," including both formal and unspoken rules which guide how team members work together. These business rules supplement knowledge of the plan, which focuses primarily on the tasks to be performed. Examples of business rules are the team's "netiquette," rules for communicating and critiquing each other, rules under which team members should ask for or offer help, rules on whom to invite to meetings, policies on seeking outside input, rules for when to speak at meetings, policies of when to seek consensus and when to "agree to disagree," and processes for teams to discuss and change the team composition, plan, and interaction methods. Team members also need to know effective ways to communicate with each other, taking into account each person's background, ability to understand meaning when expressed different ways, and possible misinterpretations of requests.

This enabler divides team business rules into two broad categories as shown below: team etiquette and team policies:

a. *Team etiquette*
   i. Agreed ways to criticize, provide feedback on work performance
   ii. Agreed methods for resolving conflicts
   iii. Agreed ways to discuss irritations about other's behavior or personality

b. *Team policies*
   i. Resolving work priorities
   ii. Asking for / offering / agreeing to provide help
   iii. Asking for / offering / relaying information
   iv. Critiquing or commenting on others' work
   v. Way team makes different kinds of decisions

vi. Seeking / accepting additional perspectives

vii. Identifying, discussing, and resolving disagreements

viii. Dealing with strong / disruptive personalities

ix. Consensus development; negotiation methods

This enabler stresses that all team members need to know what the team's business rules are. In addition, they also need to know what kinds of rules are effective. For example, a business rule that encourages team members to critique prevailing team views is essential for countering "groupthink," the team mind set that contributed to the Bay of Pigs fiasco. Management books on teamwork and the popular press discuss effective business rules at length. For instance, they stress the importance of using "I" statements when discussing behavioral problems, or of paraphrasing what others say to ensure that one is understanding another's views. The recommendations in the Collaboration Advizor™ Tool, described in Section 3, include many of the most important methods for effective team member interaction.

Importance of knowing team interaction methods. Teams whose members do not know their team's business rules often cannot work together effectively. They fail to send information to those who need it, and do not help those needing support. Team members may fail to speak up at meetings or talk too much, dominating the discussion. Team members may act in ways that seem rude or may slight other team members, causing team members to be irritated or angry with one another. Perhaps most significantly, teams may fail to seek and integrate needed perspectives, both from within the team and from outside contributors.

Knowledge of these conventions matters the most in teams and tasks that require extensive team member interaction. This is particularly true when team goals and methods of approach are unclear and require team brainstorming. It is also more of a factor when some team tasks cannot be accomplished by people working individually but require extensive dialog and mutual critiquing. It is further exacerbated by teams with some strong personalities or teams with weak leadership.

Team interaction methods are the most culturally sensitive of the knowledge enablers. Business rules that work well in one culture may be inappropriate for another. For example, in some cultures team members are considered insubordinate if they criticize the leader's viewpoints, while in other cultures such behavior is regarded to be the duty of every team member. Because the behavioral norms of different cultures often vary considerably, people working in multicultural teams should pay special attention to discussing, establishing, and documenting the team's rules.

Obtaining understanding of team interaction methods. Default team interaction methods are inherited from the norms of a culture for a nation, profession, or type of organization. Because not everyone will necessarily have the same defaults, it is important for team members to discuss and publicize its rules. Understanding of team business rules increases as team members work together, especially if the team provides

feedback when members do not adhere to their interaction protocols. Team leaders who explain and reinforce these protocols and group discussions that address this subject help provide the needed understanding.

There are several factors that impede team members from understanding their team's business rules. The biggest factor is that the team never made them explicit. Thus, teams need to make clear their rules about when to ask for or offer help or information, its rules about critiquing and commenting on others' work, its policies that facilitate hearing all people and perspectives at meetings, and its policies for handling disagreements. Sometimes tools, such as brainstorming software, can support team member interaction policies, and lack of such tools can increase the risk that desired methods will not be used. Many of the impediments to knowing team business rules are the same on the ones that impede team members from knowing one another. These include newly formed teams, team members with little experience working together, or teams whose members join or leave during the collaborative process.

Evaluating understanding of interaction methods. Evaluators can ask team members what the business rules are, including team policies about who should be cc'd, about "flaming", and about when help or information should be sought or volunteered. Team members can also be asked whether they have difficulty understanding the rules, and if they noticed people not following rules.

Significant behavioral indicators are team meetings that people complain about, as for instance can occur if team members believe that a few people dominate the meeting, that some people cannot be heard or are not listened to, or that meetings wander off topic. Other indicators are complaints about not being informed or cc'd on e-mail, or complaints about receiving too many irrelevant e-mails. Still other indicators are people not receiving or offering needed help or information, people not being informed about the availability of information that they need, or teams ignoring outsider views because they come from outsiders and do not support current team positions.

Enabler 6: Task skills—knowing how to do one's assigned work

Knowledge. This enabler is the knowledge team members need in order to perform their tasks. It includes not only task performance knowledge itself, but also knowledge about how to use tools that can help with the tasks, how to find information, and how to locate and engage people who can help them. Details of needed knowledge include:

a. *Mechanics of performing tasks*
   i. Procedures for executing tasks
   ii. "Professional" standards for ensuring quality / avoiding errors

b. *Using support tools*
   i. The capabilities of tools able to support task needs
   ii. The mechanics of using tools

c. *Finding and getting needed information*

i. Means of identifying sources of information from documents / databases

ii. Sources of information from documents / databases

iii. Means of obtaining information from documents / databases

d. *Getting help*

i. Identifying when help and/or information from people is needed

ii. Means of identifying who can provide help / information

iii. Who can provide help / information

iv. Means of obtaining the help / information from others

Importance of task knowledge. No matter how well team members understand goals, the plan, each other, or business rules, teams cannot be successful if the individual team members lack the skills and knowledge to carry out their tasks. Not knowing how to perform one's task and being unable to obtain adequate help can cause poor quality products, tasks not being done, team members not being supported, unaddressed risks, and missed opportunities. Even when the team member can get the help he needs, poor task knowledge can delay work and overload help providers.

Task knowledge is especially important when schedules are so tight that there is not enough time to redo work, and when there are information choke-points, poor team backup, and poor communications with backup providers. Information choke points block information if a single person fails to convey critical information. Poor team backup can arise if the team has not provided for backup expertise, if team interaction protocols are unclear about when help can be offered or requested, or if team members cannot monitor team member activities, tasks, or the environment as needed to see that help is needed. Poor external communications can reduce the efficiency with which help can be provided. This can occur, for example, when team members cannot share graphical information with each other.

Obtaining task knowledge. The team can obtain the needed task knowledge by selecting team members with the required skills, by training, from decision aids and other task support tools, and by providing coaching and support during task performance. Team members can ask for help if they know other team members' capabilities, if they can contact others, and if team processes permit help to be asked for.

The principal factor that increases the likelihood that team members will need help is team members with inadequate training or experience for the jobs to which they are assigned. This can occur, for example, when people with the desired experience are not available, or when team members are assigned to tasks based on title rather than skill. Other factors are inadequate task aids and information finding tools, lack of subject matter experts to provide expertise backup, or lack of communication tools that help

people explain ideas graphically in real time. Anomalous unanticipated events increase the risk that team members will encounter hard problems that they cannot handle.

Evaluating task knowledge. Task knowledge is routinely evaluated in school, when students take tests that ask them to solve the kinds of problems that they are supposed to know how to handle. Test questions can also ask about the theory behind carrying out a task, about the tools and information that helps them do a task, about how to find needed information from databases, documents, or people, or about the consequences of doing a job in a particular way.

There are many behavioral indicators of not knowing how to do one's job, some of which are obvious. The immediate effect of poor task understanding is team members getting confused and not knowing what to do when anomalies arise. This ultimately leads to poorly performed tasks or poor team products that may be rejected and must be redone. Less direct indicators are team members complaining that they need to do another's work, team members failing to relay or report critical information, problems discovered late, the team being surprised by events, and the team often having to react to problems rather than anticipating and preventing them.

Enabler 7: Activity awareness: knowing and assessing what others are doing

Knowledge. Activity awareness is the first of the six "real time" enablers. These enablers are the knowledge and understandings about what is happening moment to moment. The activity awareness enabler is the knowledge and understandings about what others on the team are doing, moment by moment. It includes knowing what they're working on, knowing how busy and engaged they are, and knowing whether that work is what the person should be doing. Specifically, activity awareness includes:

a. What others are doing
- i. What others are working on

b. Their level of engagement and busyness
- i. How busy they are
- ii. How engaged they are

c. Whether it is work that the team member should be doing (including for self)
- i. Extent work is called for by plan
- ii. Extent that work is still necessary given current situation, others' activities, work status, and likely plan adjustments
- iii. Extent that work supports team goals

Importance of activity awareness. Activity awareness encompasses knowing how busy others are, if they are getting behind or over their heads, and if they need help with their workload. Teams whose members are not aware of others' activities cannot notice and

help correct actions that are contrary to the plan or plan goals, which are inappropriate to the situation, or which may conflict with the actions and goals of other team members. Without this activity awareness, it is more difficult to know when to offer help or information, when to alter support actions, or when to clarify the plan, goals, or desired behaviors.

The need for activity awareness is increased by any factor that increases the chance that isolated team members may take wrong, unneeded, or conflicting actions. For example, activity awareness is important whenever goals are unclear or whenever tasks depend on the situation, especially in uncertain environments. It is also important when team members need to synchronize with one another, when they are performing highly dependent tasks, and when team members are inexperienced and may need help.

Ways to obtain activity awareness. The most direct way for team members to know what each other are doing is for team members to watch each other work. The ability to do this is often called "team transparency." Team members can also infer others' activities in several ways in addition to team transparency. Knowledge of others' activities can be obtained by communicating about tasks. Voice communication is especially helpful for activity awareness, since tone of voice can convey level of busyness or engagement. Activity awareness can also be obtained less directly by monitoring task progress or product development or by watching the external situation to see the impact of team member actions on the situation.

Although poor activity awareness can arise in co-located teams, it is primarily a problem in physically distributed, virtual teams. It is exacerbated when team members cannot see one another do their jobs, when it is inconvenient for people to discuss their activities with others, when they cannot view each other's work products as they progress, when they cannot share work products visually or graphically, and when it is hard to see quickly the changes people make to either the situation or to team products.

Evaluating activity awareness. The most direct way of ascertaining activity awareness is to ask each team member what other team members are doing and how busy and engaged they are, and then comparing this answer with what the other team members say they are actually doing and how busy and engaged they are.

There are many behavioral indicators of poor activity awareness. These indicators are usually ambiguous, for they can also be caused by poor knowledge of business rules or of the plan. One behavioral indicator is failure to notify teammates that they are carrying out tasks that are no longer needed or that they are failing to start tasks that are needed. Other indicators are team members that keep asking what others are doing, that take actions or initiate communications that interfere with others' current activities, or that do not offer to help others who are overworked.

Enabler 8: Understanding the external situation

Knowledge. The external environment includes all of the events and actors external to the team which may impact the team's work. In military operations, for example, it

includes the actions of the adversary. In business, it may include the actions of competitors and the preferences of customers. Understanding the external situation includes knowing who the significant players are and knowing their behaviors, strengths, weaknesses, objectives, and plans. It includes an understanding of how events lead to changes in the situation. It also includes understanding the information that supports these assessments, the uncertainty in the assessment, and the information which if available would resolve these uncertainties.

This enabler includes all of the items listed below. It includes both the "surface" understanding and "deeper" understanding. The former includes the current location, identity, status, and capabilities of people and organizations external to the team. It also includes understanding the principal uncertainties. The deeper understanding is an underlying model of the situation that explains how the situation "works"—what causes lead to what effects under what conditions. This logic allows people to infer adversary / competitor goals, intent, and plans, to project into the future, to identify opportunities and risks, and to create evidence audit trail / reliability / arguments for and against various team member views on what is happening. Specific elements on this enabler are:

a. *What is happening*

- i. The information / data that's been collected
- ii. Completeness / reliability / accuracy / pedigree of collected data
- iii. State of environment (e.g., weather, geography)
- iv. State of "neutrals" including activities and attitudes
- v. State of adversaries / competitors, including locations, status, actions, and capabilities
- vi Uncertainties of above assessments

b. *What is significant to the team and why*

- i. Collected information that could cue needed team responses
- ii Aspects of "what is happening" that could cue needed team responses
- iii. Opportunities for team
- iv. Threats and risks

c. *Projecting future events*

- i. Envelope of possible future events given assessments
- ii. Factors that influence which of these possible futures will occur
- iii. Observables / signs that these possibilities are more likely or will occur

iv. Impact of future events / conditions if any of these events does occur

Importance of understanding the external situation. Knowing the external situation is not always important, for the work of some teams does not depend on events outside of the team. However, sometimes knowing the situation is the key issue facing a team, and the team cannot succeed without certain critical information. For example, the military cannot attack an adversary if they do not know where he is, the police cannot arrest a suspect if they cannot find him, and an organization cannot intelligently invest in new products if they do not know what products their competitors will be introducing.

In general, teams that do not understand the external situation cannot intelligently modify their actions in response to a changing environment, cannot select appropriate actions, and cannot proactively adapt their posture in anticipation of changes. In addition, team members cannot use their knowledge of the circumstances under which various team members need help if they cannot anticipate these circumstances. Without understanding information requirements, team members will be unlikely to seek needed information.

Understanding situation uncertainties is often a very important and somewhat overlooked part of understanding the situation. People who assume that the situation is a particular way even though the data are ambiguous risk being blindsided when the situation turns out to be some other way.

Obtaining an understanding of the external situation. Understanding of the external situation is often difficult. In the military, entire organizations are dedicated to collecting and analyzing data about the situation, and then summarizing and depicting the results. Good situation summaries and depictions can help everyone on the team understand the situation. When the tasks demand it, these depictions should convey situation uncertainties and data reliability so that the team will not be blindsided by predictable but overlooked events. These depictions should facilitate drill down to supporting data, so that the team can review the evidence for various beliefs about what is happening. Situation depictions may convey or support inferences about situation actors' capabilities, goals, and plans.

Teams have a much harder time understanding the situation when these situation summaries are not available. In such cases, they may need to review the situation data itself, or distill a situation understanding from situation reports and messages.

There are many factors that impede a good situation understanding. Critical information may be hard to obtain, external monitors and sensors to collect the needed information may be inadequate, or the people who analyze the situation data may not have the experience to perform well. Sometimes good information cannot be collected because the team does not know what information it needs. Once the situation data are collected, team members may be unable to take full advantage of it. They may be unable to access it, to "connect the dots" in order to make critical inferences, to depict the results so that all team members can relate the situation to their tasks, or to keep it current.

Evaluating situation understanding. Questions intended to determine the extent that people understand the situation should address both the surface and deeper understandings. Basic questions ask about the specifics of the external environment; e.g., the location and identity of the adversary forces in military operations. Other questions ask about the significance of these specifics, including opportunities and risks, factors responsible for the observed situation, recognition of those factors that might impact planned actions, and understanding of how causes lead to effects. In uncertain situations, questionnaires can ask about the different situation possibilities and the evidence supporting these different possibilities.

A poor understanding of the situation manifests itself in a variety of team behaviors. One can infer that a team understands the situation when they take actions needed to respond to situation changes. These actions include modifying their plan, adjusting the team, providing alerts, or offering help. Conversely, one may infer that the team does not understand the situation if they do not modify their plan after a change in the situation that calls for it, if they are blindsided by an unexpected events, and if team members keep asking for situation updates.

Enabler 9: Current task assessment

Knowledge. Task assessment is knowing what tasks are being worked on and by whom, knowing the status of tasks and their current degree of success, projecting task accomplishment, and understanding the help—advice, information, and resources—that the task needs. It includes both a self-assessment by individuals of the tasks that individual is responsible for, and assessment of other tasks that that individual should monitor. Task assessment may be the biggest contributor to the "glue" that holds the team together, for it enables people to adjust their work to fit in with what others are doing.

This enabler complements activity awareness, Enabler 7. That enabler focused on the activities and status of individual team members whereas this enabler focuses on task progress, regardless of who is responsible for performing the task.

As detailed below, task assessment includes an assessment of actual task status, of task progress, and of task needs. Task status is the "surface" knowledge of task assessment. It is just the facts of what is happening. Understanding task progress requires some deeper understanding because people need to know what progress looks like in order to evaluate it. They often need to know the plan's specification of the progress to be achieved at various times, the indicators of progress, and the task's measures of effectiveness, all knowledge covered by the second enabler. Understanding of task needs requires knowledge of dependencies—knowing the consequences should various required resources and information be unavailable.

Task assessment addresses many different kinds of things, depending on the nature of the task. For information dissemination tasks, it evaluates the effectiveness of information gathering and distribution, including whether needed information is being obtained and properly shared. In brainstorming tasks, it assesses the extent to which needed perspectives are being obtained and appropriately considered. It is also concerned with

general team processes. It evaluates the team’s performance in critiquing, negotiating, discovering differences, and integrating concepts. It evaluates the extent to which the team is coordinating and synchronizing appropriately.

The following details the many different kinds of knowledge and understandings included in this enabler.

a. *Task status*
   i. Tasks currently being executed

   ii. Specific task issues being addressed

   iii. People working on the task

   iv. Resources available or employed in supporting tasks

   v. Information available or used in tasks

b. *Signs of problems, and if a task is on track*
   i. If task not as far along as it should be

   ii. If task process not addressing needed issues

   iii. If needed information / perspectives not available, not being solicited, or not being used

   iv. If needed resources or personnel not adequate

c. *Task needs: advice, information, resources, labor*
   i. Product critique required

   ii. Specific additional information / perspectives needed

   iii. Additional resources / labor needed

Importance of task assessment. Task assessment is essential for the team’s staying on track, and for making the needed adjustments should it get off track. Without task assessment, team members are at great risk of failing to obtain needed information and resources, failing to share information appropriately, and failing to support one another. Product-focused teams may fail to gather, share, critique, and integrate needed perspectives, sometimes leading to very poor quality decisions associated with Groupthink. Action-focused teams may synchronize and coordinate poorly, missing hand-offs, failing to mass effects, failing to prepare the ground for each other, and failing to cue one another to needed actions. In addition, without these task assessments, it is difficult for the team to hold itself and its individual members accountable, thereby decreasing team member motivation to perform well.

Anything that increases the need for team adaptability, its ability to respond to changing circumstances, increases the importance of being able to measure task progress. Thus, plans that must work in uncertain situations, particularly those in which the team must respond to an adversary or competitor, increase the importance of measuring progress. Anything that increases the need for synchronization, in which one part of the plan cannot be successful until another part of the plan is, also increases this need. Anything that increases the importance of team and individual accountability, such as a skeptical management who wants to see proof of progress, increases the need for progress assessment. Poor understanding of task status is also important when some team members are inexperienced, when the process for performing the task is unclear, or when the team is working on novel or very high stakes issues that need to consider a full range of perspectives.

Obtaining an understanding of task status and needs. There are two parts of understanding task status: (1) determining the status of tasks and task processes and (2) evaluating the adequacy of this status for achieving task objectives. People can determine task status from individual team member reports, by observing the creation of task products or reviewing their status, and by observing the impact of team activities on the environment. Understanding task status is facilitated if team members can directly inspect the product as it develops, when the product is being created in a shared environment, by situation depictions if the “product” is manifested as visible changes to the external environment, and by automatic notification of product status changes, as may occur for example using bulletin boards or notification subscription lists. Evaluating task adequacy depends on applying the task measures of effectiveness to observed effects. Experience in multiple tasks and cross training helps team members carry out such evaluations. Documents and discussions about the plan, goals and progress indicators, task resource and information requirements, and team interaction processes help team members evaluate task status and progress.

There are many possible impediments to understanding task status and progress. These include anything that makes it hard to see what others are doing, anything that makes it hard to see the product as its being developed, anything that makes it hard to judge whether progress is sufficient for the task to be accomplished successfully on time, and anything that makes it hard to know whether the team will get the resources and information it needs. Thus, physically distributed teams, a lack of a shared environment for creating team products, or a lack of automatic notification of product status changes impedes assessment of task status. Factors that impede evaluating whether the task is on track include poor understanding of the plan and a poor understanding of progress indicators. For example, it is difficult to judge the adequacy of progress if a team has no schedule, no milestones, and no product quality metrics. Teams whose members are not trained to perform other’s tasks and teams whose members have little experience comparing observed actions with actions called for by the plan impede task understanding.

Evaluating how well team members understand task status and progress. Evaluators can ask team members about the status of various tasks, about processes for successful accomplishment of these tasks, about the information and resources required at various

points in task development, about criteria for evaluating task progress, and about the current status of the task with respect to the status expected by the plan at that point in time.

In addition to asking team members questions, evaluators can infer their understanding of task status and progress both from the quality of the task monitoring infrastructure as well as from team behaviors. That is, if there is no apparent way for team members to get information about task status, then one can infer their understanding is unlikely to be very good. If they behave in ways inconsistent with a good understanding of task progress, then their understanding is unlikely to be good. Infrastructure indicators are the ease of reviewing the task products, the quality of information about task status, the availability of written progress metrics, and the types of tools for tracking progress. Behavioral indicators of task status understanding are coordination actions taken to adjust for rate of progress, team members critiquing or enriching team products at appropriate progress points, team members providing timely notification of important events to other team members, and team members requesting or offering additional information or help to improve products. Indicators of poor team status understanding are people not being informed when they are is doing something contrary to the plan, people carrying out actions that are no longer needed, team members not finishing activities at the right time or right manner to support needs of another person, and team members consuming or reserving critical resources needed by others.

Enabler 10: Mutual understanding

Knowledge. Mutual understanding is being aware of what other team members' views and understandings are, knowing the reasons for these views, and as a result, knowing when team members agree or disagree. It includes the extent to which team members are aware of where they agree or disagree about team goals, team progress, the external situation, and all the other knowledge enablers. Mutual understanding includes:

a. *Transfer of meaning—how others understand input*
    i. Extent that others understand what is being said
    ii. Possible alternative understandings
b. *Where others agree or disagree and why*
    i. Issues on which people disagree
    ii. Reasons for disagreement; information, values, world models
    iii. Importance of addressing / resolving disagreement

Mutual understanding focuses on teammates' current beliefs about team issues. It begins with an assessment of "transfer of meaning," knowing whether one is being understood or knowing how other people are or might be interpreting what is being said or being observed. Though this assessment builds on "knowing others" (Enabler 4), it differs from

that enabler because “knowing others” is concerned with the longer term, relatively stable beliefs and values of other people and not on their immediate perceptions.

Knowing the reasons for disagreements is an important part of this enabler. There are usually only three basic reasons why people disagree, and part of the knowledge for this enabler is knowing which reasons apply in a particular case. The first reason is that people have different information. The second is that they have different values or goals. The third is that they have different models of how the world works; e.g., what causes lead to what effects.

Mutual understanding also includes understanding team consensus positions; e.g., knowing what the “team view” is. It is concerned with knowing the extent to which team members are aware of each others’ level of commitment to agreed actions.

Importance of mutual understanding. First, it is important to distinguish between team members knowing what others’ views are and everyone on the team having the same view. In teamwork and collaboration, it is usually not necessary that all team members agree on everything. In fact, it is sometimes unhealthy and even dangerous. Teams that think it is important for everyone to agree can suppress unpopular viewpoints, leading to very poor team performance, as happened in the Bay of Pigs. Even consensus, which is often essential for a team to achieve, does not require that everyone on the team share the same view. Consensus just means that everyone agrees to support a particular position, not that everyone agrees with it.

A team in which team members do not share the same views and do not know that they do not is prone to many difficulties. When they do not share the same understanding of goals, then team members may take actions that surprise other team members and that seem counterproductive to them. One team member may, for example, fail to provide the support that another expects. Lack of mutual understanding can seriously undermine coordination, for the different team members may have different expectations about the type of coordination required and the circumstances under which it is done. Team members who think they understand one another but do not may never have the discussions needed to resolve these differences of understanding. Team members who disagree and who do not know the basis for the disagreements may never be able to reach consensus.

Mutual understanding is especially important whenever tasks require close coordination and synchronization, or whenever the team’s mission requires that the team reach consensus.

Obtaining mutual understanding. Access to common information about goals, plans, team status, task status, situation status, and each other improves mutual understanding. Mutual understanding also grows from long-term background knowledge of each other. People who have worked together for a long time learn how each other are likely to interpret goals, situations, and plans, and are likely to know the criteria each other uses to decide what to do. If people have not worked together in the past, then some of this long-term knowledge can still be obtained if they are familiar with each other’s backgrounds.

People will build on and strengthen this understanding of others if they can watch other people work, review their products, and see how they behave under various circumstances. The ability to communicate, especially communication that conveys non-verbal cues, helps people assess the extent that others are understanding and agreeing with what is being said. Team leaders able to detect differences in understanding can be especially helpful in supporting group processes to achieve needed alignments.

The major factor impeding mutual understanding is physical separation of team members and, for distributed teams, an environment in which simultaneous promulgation of information is difficult, or in which it is difficult to communicate such non-verbal cues as tone of voice, facial expressions, or body language in real time. Interaction processes where team members do not acknowledge lack of agreement seriously undermine mutual understanding. In addition, anything that impedes team members from individually understanding goals, plans, dependencies, each other, interactions, activities, the external situation, and task status also hinders mutual understanding. Thus, the lack of common operational pictures and common views of the plan and plan status hinders mutual understanding. New teams, especially when the team members have not worked together before, have different job specialties, or come from different cultures greatly complicates mutual understanding.

Evaluating mutual understanding. The basic way to evaluate mutual understanding is to ask one person what another's views are, and then to compare those answers with the views that other person says he has.

Behavioral indicators of poor mutual understanding are coordination and synchronization failures, team members talking past one another, team members failing to reach consensus, team members taking actions that surprise each other, team members who keep repeating the same point, and people who fail to probe for the reasons of discovered disagreements.

Enabler 11: Plan assessment

Knowledge. Plan assessment is an estimate of whether the current team, processes, plans, and resources will still enable the team to achieve its objectives. It builds on and integrates activity awareness, task assessment, understanding of the external situation, and mutual understanding. Unlike a task assessment, which focuses on how well individual tasks are progressing, plan assessment considers all current factors and projections into the future to estimate the need for plan adjustments.

To assess whether the plan will still work, team members need to project the future status of the environment, of resources and information. Then they need to understand task and situation dependencies in order to estimate how these projections impact plan viability.

Thus, as shown below, plan assessment has two steps. In the first, team members review what the plan needs in order to succeed, and then projects the extent to which these success conditions can be met. In the second step, team members estimate the prospects

for achieving future tasks and goals given the prospects for meeting the success conditions. Specific plan assessment includes:

a. *Projections of resource, people, and information availability*
    i. Resources, people, information, situation conditions needed for plan to still work

    ii. Possible future status / availability or resources, people, information, and situation conditions

    iii. Factors that can impact this future status

b. *Plan prospects and danger points*
    i. Future tasks at risk given projections

    ii. Expected outcome of plan execution / goal achievement given risk projections

Importance of plan assessment. Teams that cannot determine whether or not their plan can still work cannot know when they need to make changes in order to attain their objectives.

Anything that requires team agility increases the importance of plan assessment. Plans that depend on the situation, especially when plan success depends on the actions of a competitor or adversary, increase the importance of assessing the plan’s viability. Frequent plan assessment is also important when situation and task dependencies are uncertain and when it is difficult to project the consequences of actions, as occurs in complex or unfamiliar environments. It is also important when some team members are inexperienced and may have difficulty carrying out their responsibilities.

Obtaining the knowledge needed to assess the plan. People get the needed status understandings from the team activity, situation, and task assessments and from the assessment of mutual understandings. Team members understand the potential problems that might arise if the plan is not changed by projecting the plan and situation status into the future. This normally requires understanding of the dependencies—how actions lead to effects—between the situation, plan requirements, and the achievement of tasks and goals.

Anything that impedes obtaining situation, resource, task, and mutual understandings also impedes plan assessment. Thus, impediments to team members knowing one another, to activity awareness, and to understanding goals, dependencies, task status, degree of mutual understanding, and goals impedes plan assessment. For plans that depend on the external situation, any impediment to projecting the situation can especially hinder plan assessment. Factors that make situation projections hard are complicated situation-dependent plans in which it is difficult to predict end results from current status, unfamiliar situations, inadequate information collection means, and an active adversary or competitor.

Evaluating plan assessment. Questions can address team members' assessments of whether the plan will succeed, their assessment of the factors that may jeopardize plan success, and their reasons for believing that these factors may emerge.

Behavioral indicators of poor plan assessment include the team focusing on the wrong issues and failing to make needed adjustments to the plan after reported critical situation events. These failures can be difficult to note at the time they occur, but can be inferred later when small problems become big ones, or when the plan runs into significant unanticipated problems such as inadequate resources, information, or adversary/ competitor actions.

Enabler 12: Understanding of decision drivers

Knowledge. People continually make decisions when they collaborate. They decide what information to collect and share, how to prioritize their tasks, whether they need to discuss adjusting the plan, and what they should do next. All of these decisions build on an understanding of task and plan progress—on an assessment of current team and individual successes and problems and on the reasons for these successes and problems. Those decisions needed for team agility also rely on judgments of the extent to which current team, processes, plans, and resources will enable the team to achieve its goals, and if not, why. Ultimately, the success of the team depends on the quality of the decisions that team members make.

The list below describes the knowledge that people need to make good decisions, dividing this knowledge into four major categories. The first is a list of factors that decisionmakers need to consider. Perhaps most important of these is knowledge of the "success / appropriateness" conditions for each action. This knowledge is the basis of expert "recognition primed" decisionmaking[15] where experts know the situation properties an action needs to succeed and so can make a good decision by checking whether or not the situation has these properties. The second concerns the deadline for a decision. This deadline knowledge often takes into account more than just the time on a plan schedule. It can also be driven by the time remaining for the needed situation properties to stay in place or the preparation time for implementing the decision. The third category is decisionmaking under uncertainty—an important area that many people are not familiar with. The entries list some of the different techniques for handling uncertainty, from acquiring more information to shaping the environment to diminishing the uncertainty. The fourth category is involving the right people, to include consulting people with special needed expertise or negotiating with stakeholders who need to support the decision.

a. *The right factors to consider*
   i. Goals and objectives of all stakeholders

   ii. Potential actions able to address a problem and address goals

[15] Klein, 1998; Noble, 1989.

iii. Prior agreements and understandings on possible actions

iv. Action success / appropriateness conditions for each action

v. Consequences / impact if success / appropriateness conditions not fully satisfied

b. Deadline for decision

i. Deadline according to planned schedule

ii. Time in which possible action stays feasible

iii. Time required to implement decision

c. Methods for managing uncertainty

i. Level of uncertainty

ii. Risk from uncertainty—consequences from possible unfavorable circumstances

iii. Information required to reduce uncertainty, and means / feasibility of obtaining information (including actions to smoke out adversary capabilities and intent)

iv. Pros and cons of waiting for situation to clarify before deciding

v. Finding an action that hedges for all uncertainties

vi Specifying contingency plans that hedge for uncertainties

vii. "Shaping environment" to reduce uncertainties

viii. Taking small incremental steps while waiting for situation to clarify

d. Involving right people

i. Stakeholders that decision could affects

ii. People with information / knowledge useful for decision

iii. People to be consulted / involved according to team policy

This list of factors does not presume any particular mode of decisionmaking. It does not assume, for example, that people will use the rational "multi-attribute utility" decision making in which they generate alternatives, project outcomes, rate desirability of each outcome, and then pick the most desirable. Nor does it assume that only experts are making decisions in familiar circumstances.

Importance of good decisionmaking. Poor decisions can lead to ineffective task performance, poor coordination, poor product quality, and mission failure.

Good decisionmaking is always important. It is especially important when stakes are high, actions cannot be reversed, and the team faces an intelligent adversary able to exploit any errors.

Obtaining the understandings needed for good quality decisions. Experience is the most important factor in good decisionmaking. It enables decisionmakers to identify a good set of alternative actions, provides the knowledge about when each of these alternatives works, and helps them project the consequences of an action in a given situation.

In the absence of experience, it is hard to make good quality decisions. However, there are methods for improving the understandings needed for good decisionmaking. Discussing the decisions with team members, reviewing audit trails of past discussions pertaining to the decision, and reviewing planning documents that specify what should be done under various circumstances can help clarify what to do. Estimating the consequences of possible actions requires knowledge of task and situation dependencies, projections of future resources, information, and environment status, and estimates of the capabilities and dispositions of team members. Status projections are often obtained using forecasting methods, and estimates about team members usually arise from team members' experience working with each other or from reviewing team member backgrounds. Reviewing the plan's schedule can improve understanding of decision deadlines. Situation depictions that portray uncertainty can help people understand that uncertainty, and methods for dealing with uncertainty can be taught.

Decisionmaking is harder when it is difficult to assess plan progress and prospects. It is difficult when there are multiple stakeholders or competing goals, each of which may be unclear; when the situation is uncertain, unfamiliar or complex so that it is difficult to assess the applicability of previously adapted actions and hard to estimate the effects of alternatives; when the team faces an intelligent adversary, when team member tasks are highly dependent so that actions may impact many different team members, and when it is difficult to review the rationale for current planned actions. Decisions are also difficult when scarce resources make it hard to identify a feasible alternative, when potential highly adverse side effects make the decision risky, when tight deadlines make decision makers feel rushed, when unclear deadlines impede decisionmakers from knowing how long they can wait before deciding, and when actions are not reversible so that any decision forecloses future options.

Evaluating understanding of decision drivers. It is difficult for non-experts to evaluate how well someone understands decision drivers. Such evaluations almost always need an expert answer key. Evaluator's questions are usually most effective if linked to specific incidents encountered by the team. Questions can ask people to specify an action to be taken, the various uncertainties that this action addresses and the factors it takes into account. Evaluation of answers considers the quality of action as specified by the expert answer key. The evaluation also considers the completeness and correctness of the uncertainties and decision factors considered.

During collaboration, observers can note the actions taken under various circumstances, the situation under which the action was taken, and the consequences of the action. Decision quality can then be assessed using an expert answer key that rates the quality of the action. The decision can also be associated with the desirability of the outcome. If the team discusses uncertainties and critical factors, then these can be evaluated using the same methods employed for questionnaires. Specific indicators of low quality decisions are team's being blindsided by unexpected events (often poor decisionmaking under uncertainty), team members expressing regret over some past decisions, and problems that could have been easily fixed early, but became serious over time.

3.0 The Collaboration Advizor™ Tool

Sections 1 and 2 have described the importance of knowledge to successful collaboration and have enumerated the diverse types of knowledge needed to support team effectiveness. An important point in that discussion, made in Section 2.2.1, was how knowledge and teamwork support each other. It pointed out that not only does good teamwork depend on having the right knowledge, but acquiring the needed knowledge depends on good teamwork. Thus, getting the knowledge needed for future work depends on having the knowledge needed for current tasks. Because of this dynamic, small gaps in current knowledge can grow into big team problems. Accordingly, it is important to catch small problems in knowledge early, before they turn into big problems.

Unfortunately, because the range of knowledge needed for effective collaboration is so extensive and diverse, it is not easy for teams to be aware of everything that they need to know. Accordingly, it can be hard to recognize that one's team lacks some of this needed knowledge, to realize that the team needs to discuss some of these knowledge issues, and to identify effective ways to remedy knowledge gaps.

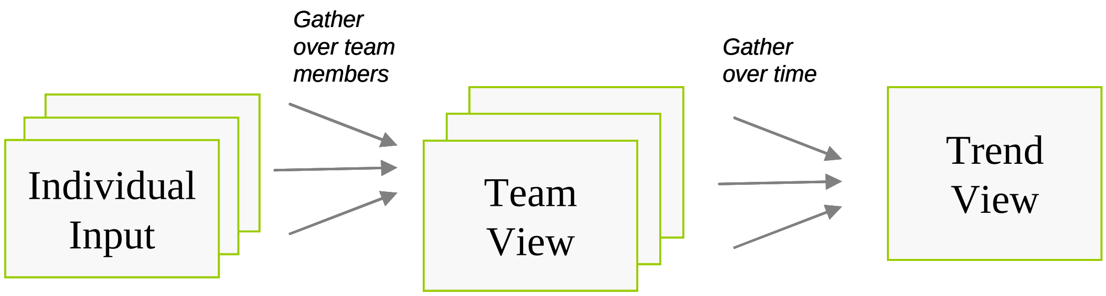
The Collaboration Advizor™ Tool addresses these problems. It helps educate team members, helping them to be aware of knowledge and understandings that are important for teams to have. It helps diagnose knowledge deficiencies, pointing out areas where team knowledge may be weak or where team members have conflicting viewpoints. It helps teams handle social issues, providing a socially safe forum for airing differences. Finally, it helps identify how to correct knowledge deficiencies. By using all of these methods, this expert system helps teams diagnose and fix small knowledge problems before they get to be big problems.

This section discusses this tool. Section 3.1 describes the tool functions, reviewing what team members experience when they use the tool and how the tool fits within the team development cycle. Section 3.2 then describes how the tool works—the organization and content of the knowledge base and some of the algorithms. Section 3.3 reviews the phases of tool development and testing, describing some of the experiences of teams who have used the tool.

3.1 Using the Collaboration Advizor™

3.1.1 Collaboration Advizor™ Modes

The Collaboration Advizor™ has three different modes (Figure 10): (1) an individual view mode where team members answer questions and explore issues independently; (2) a team view mode where the team reviews team problem areas, identifies and discusses areas of agreement and disagreement, and decides how to fix problems; and (3) a trend view mode where the team can review its progress over time. The following sections illustrate tool support in each of these areas using the story boards for the commercial tool version under development.



Individual Team Members

- Answer questions about team (anonymously if desired)
- Explores issues

Advizor Tool

- Alerts to possible problem areas
- Gives reasons for concerns
- Lists warning signs, impediments, and recommendations
- Presents additional optional questions
- Saves responses

Tool

- Consolidates individual responses
- Lists team view on areas of concern
- Shows areas of agreement and disagreement

Team

- Discusses areas of agreement and disagreement
- Reviews recommendations
- Decides on follow-on actions

Tool

- Shows changes to enablers over time
- Shows changes to impediments and symptoms over time
- Overlays key events

Team

- Reviews progress
- Decides on changes

Figure 10. The Collaboration Advizor™ Modes

In addition to these views, the tool also provides management and set-up tools. The knowledge base editor allows collaboration experts to modify the Collaboration Advizor™ knowledge base. The leader set-up form enables a team leader to tell the tool about team, task, and environment characteristics that impact the importance of various kinds of knowledge to that team. It also enables the leader to tell the tool whether the team is just getting started and is still organizing itself, or whether the team is now beyond that and is executing planned tasks. It does not ask the leader any questions about how well the leader thinks the team is doing. The leader would answer those as a member of the team during the individual mode. Both the knowledge base editor and leader set-up forms are "behind the scenes" to team members using the tool, and are not discussed further in this section.

Starting the Collaboration Tool. When a team member logs onto the collaboration tool, he or she is asked to choose the tool mode: input, team view, or trend. Users may change modes at any time when using the tool.

3.1.2 Individual Mode: Input, Diagnosis, and Exploration

Questions. In individual mode, team members answer questions about the team and explore collaboration issues. The tool provides an option for team member input to be anonymous. Most teams choose this option because they feel it promotes a more accurate diagnosis of team problems and because it helps surface issues that the team needs to discuss.

At the start of the session, the tool asks the user approximately 50 questions about the team and tasks in two phases. It tailors the questions to reflect the leader's input: whether or not the team is still in its organization phase and the a priori importance of a team's having the knowledge particular to each of the twelve knowledge categories. If the team is still organizing, then the tool omits questions about those symptoms of poor inadequate team knowledge that would not have had time to occur.

Figure 11 depicts the user interface for the first page of questions asked in the first phase. In this phase the tools asks two questions about each of the enablers. The user checks the ones that he or she feels apply to the team. The user may check neither, one, or both of the questions for each of the enablers.

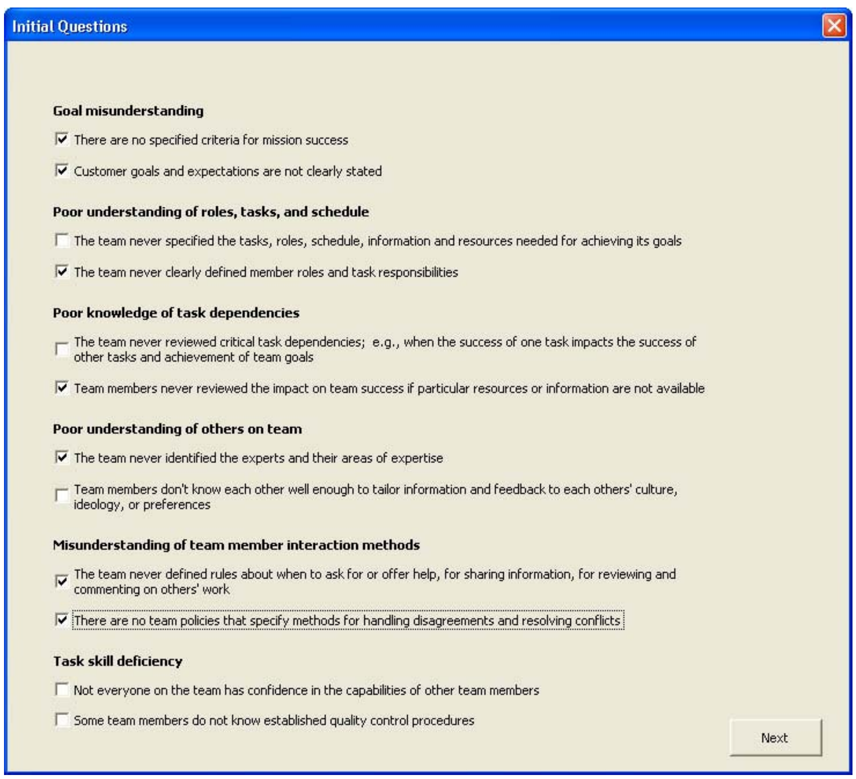


Figure 11. Advizor™ Tool Input Form

After the user answers the first 24 questions, the Advizor™ Tool picks the next set of questions. The total asked in this phase will vary from 20 to 30. The number of questions asked about each of the knowledge categories depends on the user's answers to the first

24 questions, on the input on knowledge importance from the leader form, and on the inherent benefit of team members seeing a particular question, as defined in the knowledge base. Note that because the particular questions asked in this second set depend on the answers to the first set, different team members usually see slightly different sets of questions.

When the user completes this second set of questions, he has answered all mandatory questions. Later on he may answer additional questions on team problem areas if he chooses to; however, these questions are optional. The user now moves forward to view the tool's diagnosis based on his or her answers to these questions. In individual mode, this diagnosis depends only on that person's input, in contrast to the team view mode which considers input from all team members.

Diagnosis

After the user presses the diagnosis button, the Advizor™ Tool presents the diagnosis summary screen (Figure 12). The left side of this screen summarizes the strength of team knowledge in each of the twelve enabler areas. The right half of the screen reviews some top level issues about any particular knowledge enabler that the user may wish to examine.

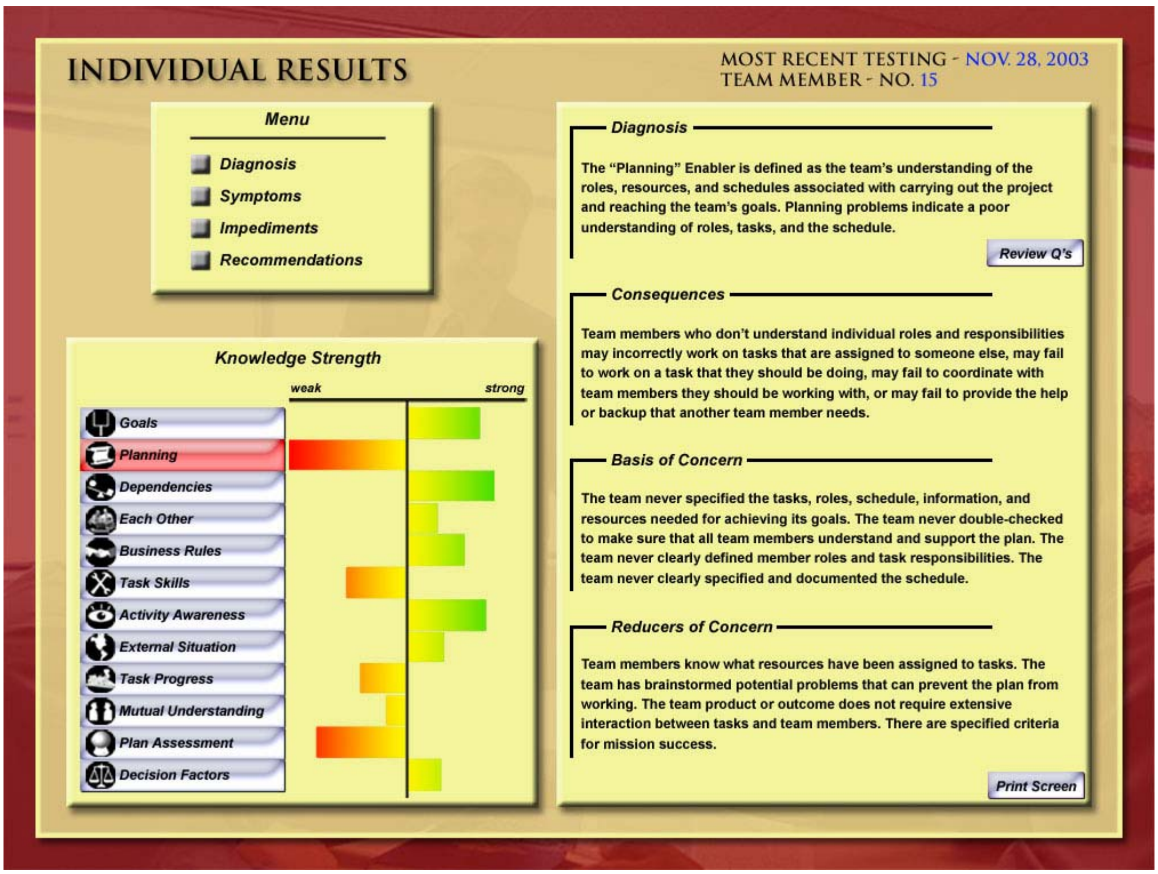


Figure 12. Diagnosis Summary in Individual Mode

In the diagram summarizing the status of knowledge in each of the enablers, the length and direction of the bar indicates the extent to which a team should be concerned about the knowledge in each enabler. Generally, a team does not need to be concerned about any bar on the right side of the center line. Teams should, however, examine further those knowledge enablers with bars on the left side of center line, especially those enablers whose termination points are colored red. To examine an enabler further, a user selects that enabler.

Once an enabler is selected, the right half of the summary interface depicts four kinds of high level information about that enabler: what that enabler is about; possible consequences of poor team knowledge in that area, and the reasons why the tool believes the team may have ("Basis of Concern") or not have ("Reducers of Concern") this problem.

The information on this screen seeks to empower the user, helping him to understand the issues and to use his own judgment about the seriousness and significance of a knowledge area that the tool identifies as problematic. Because the tool's diagnosis is based on a very small number of questions and because these questions can have many different implications for different types of teams, the tool does not assume that its diagnoses are definitive. Instead, the tool is designed to help team members to be aware of the issues, to point teams in the right direction, and to help them exercise their own judgment about what is important for this team to address.

Each kind of information on the right half of the interface is intended to support such user understanding. The first text block describes what the enabler includes, revisiting the material in the "knowledge" subsection of the enabler descriptions in Section 2.2.3. By recounting the actual content, the tool helps the user think about the specific knowledge that a team needs to have and that his team may lack. The second text block describes the consequences of a team's not having some of the needed knowledge. By reviewing the consequences, team members can decide how important it is to attend to a knowledge deficiency. If they feel that the consequences would not be very detrimental to their team given their particular circumstances, then the team may decide to defer addressing a knowledge area. They may also defer addressing the problem if they feel that the consequences are very unlikely given the specific circumstances of their team.

The last two blocks of text review the reasons for the tool's diagnoses. Like almost all expert systems, the Advizor^{TM} Tool provides the user with a rationale for its conclusions so that the user can independently evaluate whether or not the tool's diagnoses apply in any particular case. Transparency into an expert system's rationale is essential for giving users confidence in the appropriateness of the system's output, for expert systems do not know all the circumstances in which their knowledge base is not applicable. In the case of the Advizor^{TM} Tool, this rationale is a review of the questions that the user checked (third text block) or did not check (fourth one). Every question that the user checked is an indicator of some problem in one or more knowledge enablers. Every question that the user does not check is a counter-indicator of a problem in those areas. When the tool makes a diagnosis, it balances the questions that the user checked with those that it did

not check. By reviewing this rationale, users can decide the extent to which the tool's conclusions are appropriate to their team.

This summary screen is always the starting point for examining the Advizor™ Tool's diagnoses. In addition to this summary, the tool helps the user examine additional issues important to each enabler. These include the enabler impediments and symptoms, and the recommendations for improving the team's knowledge in specific areas. The tool also permits users to review and revise answers to previously asked questions. The user can access symptoms, impediments, and recommendations from the menu at the top of the left hand side of the diagnosis screen.

Impediments and Symptoms

In the Collaboration Advizor™ Tool, an *impediment* is any factor that makes acquiring needed team knowledge more difficult, and a *symptom* is any indicator that the team's knowledge is deficient. Impediments and symptoms are displayed separately on different interfaces. They share the same format.

There are many types of impediments. Many stem from the type and composition of the team: geographically dispersed teams pose significant obstacles to team members having needed activity awareness; teams with new members or rapid turnover increase the risk that team members will not know one another well enough to interact efficiently. Other impediments arise from the nature of the environment. Having an intelligent active adversary significantly complicates being able to predict whether a team's plan will still work. Still other impediments arise from the nature of infrastructure and tools. Because they lack cues from tone of voice, team members in a physically distributed team without voice communications risk being unaware when the people they are communicating with do not understand or agree with what is being said. A fourth kind of impediment, often the most common and usually the most correctable, is the failure of teams to follow recommended procedures. Not discussing goals increases the risk that team members will not know the goals; not writing a plan down increases the chances that team members will not understand the plan; not discussing team interaction rules increases the likelihood team members will not know these rules.

Figure 13 is a screen shot of the impediments for the "planning" knowledge enabler. This screen has four sections: The first is a general impediment for planning in general. The next section lists impediments which the user has already indicated apply to this team. These repeat those impediment-related questions that the user has already answered. The third section lists potential impediments that the user has already been asked about and which he indicated do not apply to the team. The fourth section lists additional impediments which the team member has not yet been asked about.

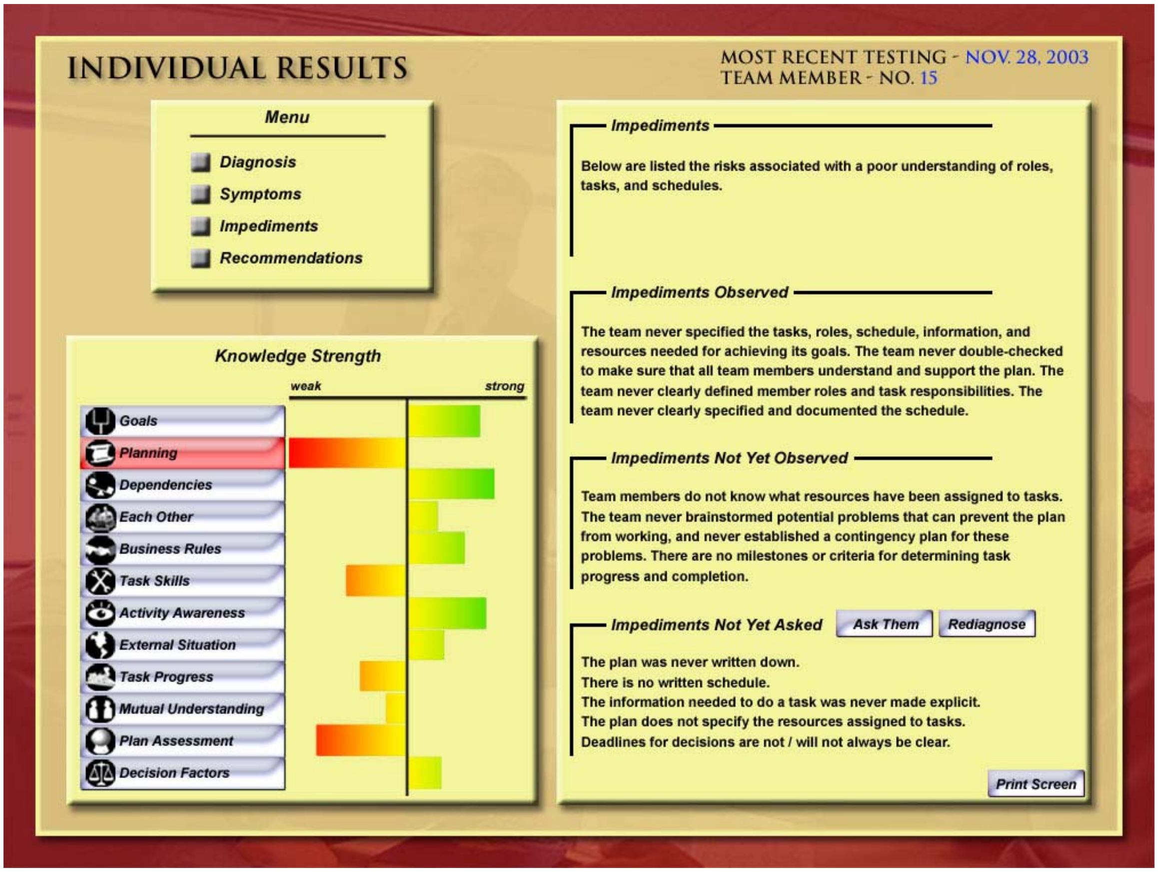


Figure 13. Tool Depiction of Team Impediments for Planning Enabler

This fourth section provides a way for team members to examine knowledge impediments in depth. The knowledge base includes over a hundred different potential impediments, but in order to reduce user workload, the tool asks the user about only between 20 and 30 of these. This section enables the user to review the rest of the impediments for that enabler. If the user chooses to do so (by pressing the “Ask Them” button), he may also indicate to the tool which of these additional impediments applies to the team. Once he does so, these answers can impact the tool’s previous diagnosis. The user can then direct the tool to update its diagnosis by pressing the “Rediagnose” button.

The tool can help the team identify means for improving its knowledge, and has a special section for doing this, the “Recommendations” facility discussed later. However, if a team is having trouble in a particular knowledge area, then looking at the team’s impediments in this area is a good way to start understanding how to fix the problem. For example, if the tool diagnosis shows that team members may not understand its plan, and if in addition the tool notes the team is at risk for not having written down the plan, then a likely way to help fix the problem is to write the plan down.

Of course, impediments by themselves do not show that a team has a deficiency in a knowledge area, only that the team is at risk for having that deficiency. Symptoms are the behaviors and conversations that reflect an actual knowledge deficiency. They are warning signs, helping team members to be aware of what they should be looking for to

determine if the team actually has a particular problem. They are like the lists of warning signs that appear in the popular press; e.g., “the ten signs that your child is depressed.”

The format of the Symptom review is identical to the Impediment review. The top section describes the general symptoms of an enabler deficiency. The second and third sections list the symptoms that the user has already been asked about and indicated that he has or has not observed these symptoms. The fourth section lists additional symptoms that the user has not yet been asked about and which the user may wish to examine. In the case of symptoms, this list is often especially useful, for it can cue the team to watch for certain critical signs of problems. As in the case of impediments, the user may indicate to the tool which of these symptoms have been observed, and after answering these questions, may instruct the tool to update its diagnosis.

Recommendations

The Advizor™ Tool makes specific suggestions about how to correct knowledge deficiencies. Figure 14 depicts a recommendation form. This form has three areas: a general recommendation for an enabler as a whole, a list of issues (impediments or symptoms), for which the tool has specific recommendations, and the list of recommendations themselves. The user can access the recommendations for any specific issue by selecting that issue.

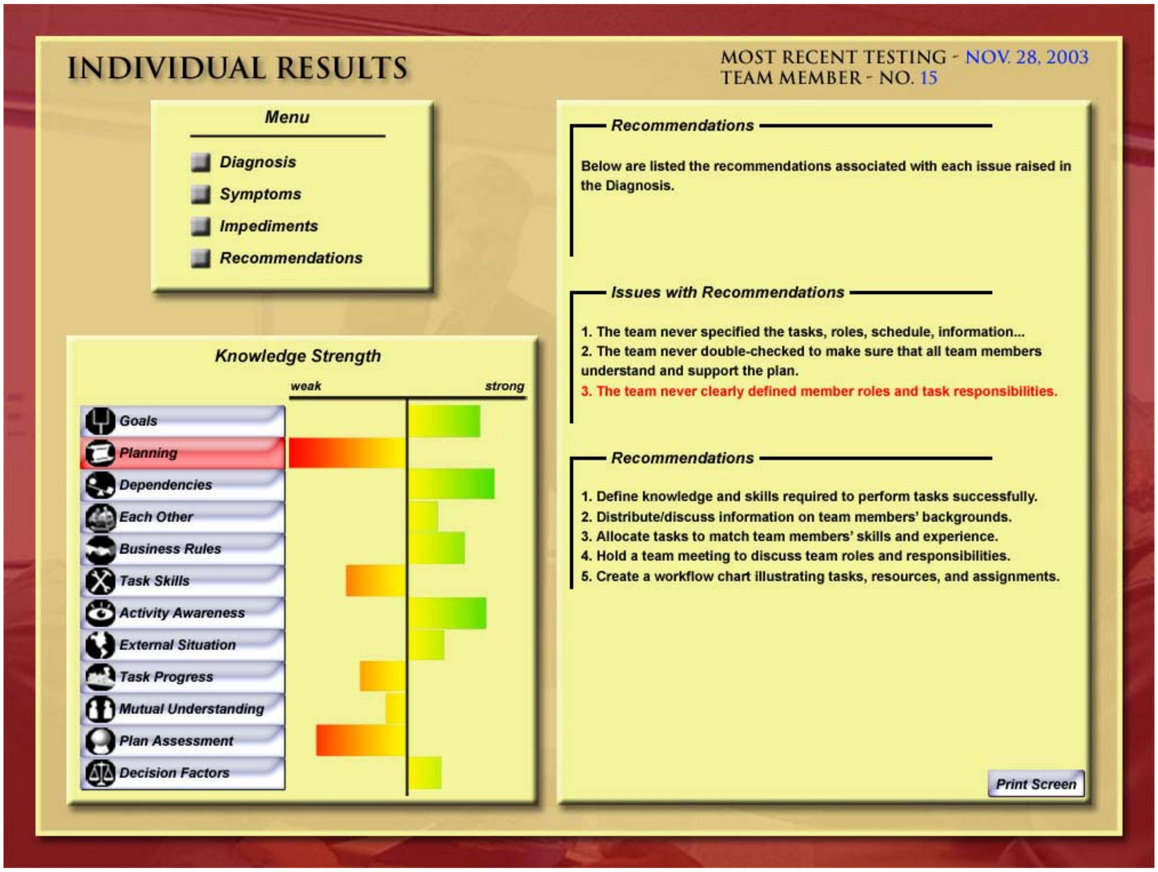


Figure 14. Recommendations

Some of the tool's recommendations are direct and obvious. If a team is at risk because it has not written down its plan, then it should consider doing that. Other recommendations are tricks that many people might not be aware of. If people are not sharing private information in some area (a symptom), then they can be motivated to do so by publicly designating them experts in that area. Some recommendations are collections of standard advice on working together. To make sure that you understand someone, paraphrase what you think is being said and ask if your paraphrase is correct. Still other recommendations provide multiple ways to handle a problem. If the team is at risk because the external situation is uncertain, then the tool will make recommendations both about how to reduce the uncertainty and also about how to cope with it.

The tool presents its list of recommendations for each issue in order. When the list is a collection of alternative approaches to a problem, then the order designates which of several ideas should be considered first. When the list is a collection of steps to be taken to correct a problem, then the order specifies the order in which the steps should be taken.

Question review and revision

The tool's individual mode permits team members to review and change the answers to their questions. When this option is selected, the tool lists all of the previously asked questions for an enabler, and shows which the user had previously designated as being true for that team. The user can then change any of his previous answers. When finished, he can direct the tool to update its diagnosis.

3.1.3 Team View Mode: Discussing Problems, Reviewing Disagreements, and Deciding on Solutions

In team view mode, the Advizor™ Tool consolidates the responses from all team members who used the tool in individual mode, and creates a new single diagnosis that reflects these responses. The team view then displays the consolidated diagnosis and the team member responses contributing to that diagnosis. These views (1) remind team members of issues important to team effectiveness; (2) help put issues that are important to discuss on the table; (3) highlight specific areas where team members agree or disagree; and (4) suggest ways to improve team knowledge.

The team view is designed to be viewed by all team members simultaneously. For co-located teams, team members can meet and view a single copy of the team view projected on a large screen. For distributed teams, team members simultaneously view the same team view at the various team member sites. This simultaneous viewing mode contrasts with the individual mode, where team members use the tool individually and asynchronously.

The team view plays two important roles in the team's development cycle. First, this view provides the "control signal" that teams need in order to know where they need to improve. Most teams need to make mid-course corrections and adapt to changing circumstances over time. The team view mode shows teams where they may need to make these corrections. Second, the team view accelerates team "hardening." This

hardening occurs as team members get experience working together. They get to know each other's capabilities and preferences, become more familiar with the team's business rules, and become more skilled at working together. The team view, by surfacing differences in understanding among team members and raising some socially sensitive areas, speeds this process.

Team View Diagnosis

The team view diagnosis integrates the answers that the team members provided in individual mode, and quantifies a "knowledge strength" (or degree of "non-concern") for each knowledge enabler. This knowledge strength increases for each team member that agrees that a question is not true for that team, and decreases for each team member that thinks the question is true for the team.

The team view diagnosis interface (Figure 15) resembles the initial diagnosis sheet for individual mode, but the numbers and items depicted reflect the views of the whole team rather than any single team member. As in the individual mode, the left half of the display shows the strength of team knowledge in each of the enabler areas and the right half shows high level information about the enabler selected for review in the left half. Team members can access more detailed information through these high level summaries.

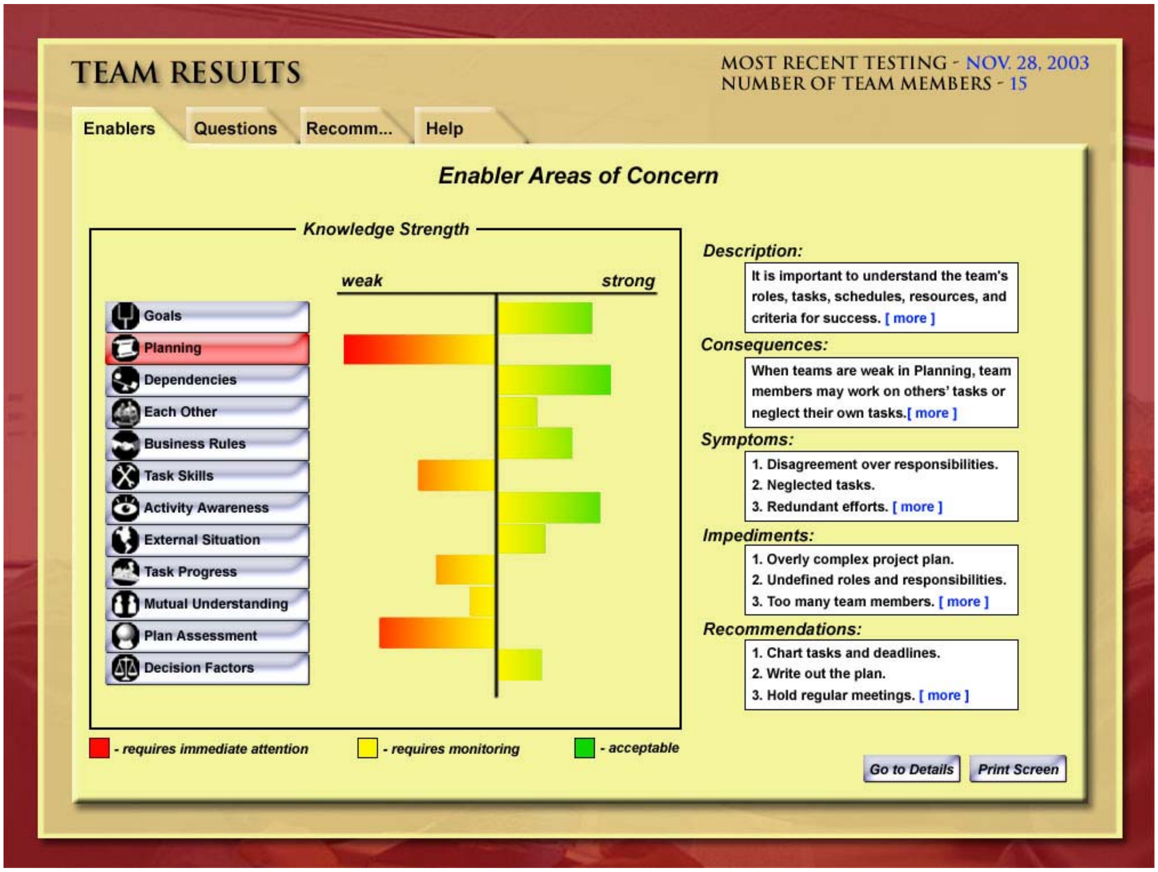


Figure 15. Team Results

As in individual mode, the team view mode seeks to leverage the experience and intelligence of the team. Thus, the team results view explains the reasons why a team could be concerned about particular knowledge areas. It explains what that knowledge area is about, the consequences of knowledge deficiencies in that area, the symptoms that indicate that that team has the problem, and the impediments that that make it more difficult for that team to obtain the knowledge. It also summarizes recommendations for dealing with the problem. The team view interface provides a very short summary for each of these areas. Clicking on the description of consequences text presents additional elaborating information. Clicking on the symptoms, impediment, or recommendations text opens the "team question" or "recommendation" displays described below.

Team Question Review

The team question review (Figure 16) helps the team review specific issues that may be important to discuss and address. This is the interface that highlights specific impediments and symptoms for that team. It also shows areas of agreement and disagreement, and provides hooks to recommendations.

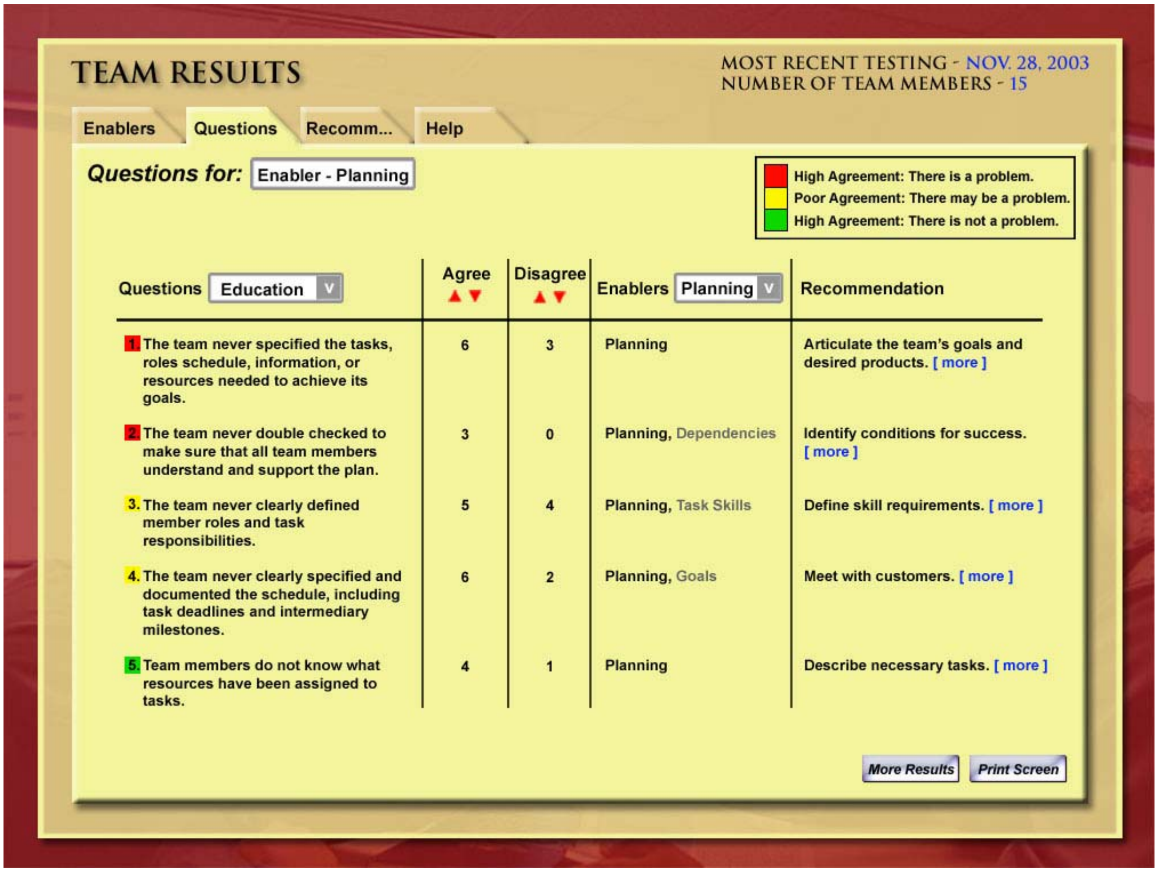


Figure 16. Question Review in Team Results

The question display shows all questions relevant to the diagnosis of one or more enablers that the team has decided to discuss. Selected enablers are listed at the top of the

form, in the block “questions for.” The team can select the enablers to be included by checking the enablers to be included in the pull down menu in the enabler columns.

The question review display has five columns: questions, agree, disagree, enablers, and recommendation. Generally, the questions column lists only those questions that at least half of the team members viewed in individual mode. Note that team members can see slightly different questions because some questions asked later depend on the answers to earlier ones and because some team members may choose to answer some questions accessed through the individual mode diagnosis form. Each question is marked by a red, yellow, or green square. Red signifies that more than 85% of team members who saw the question agreed that it was true of the team. That is, the great majority of team members who saw the question agreed that the team had that symptom or impediment. Green signifies that more than 85% of team members that saw the question think it was not true of the team; e.g., the issue is not a problem. Yellow indicates that the team members disagree on whether the issue is true of the team.

The team may select the order in which the questions are listed. If they pick “discussion,” the tool lists the questions that are most important to discuss first. These are usually questions about goals, team interaction methods, or anything that could be socially sensitive. The tool helps surface issues that might not otherwise be discussed because the tool (and not individuals on the team) actually raises the issue by asking the question. In addition, for teams that elect that the input be anonymous, individuals feel freer to answer them honestly. Examples of some of these socially sensitive questions include “some team members do not have the experience to do their tasks,” or “some team members do not know how to behave appropriately.” Disagreements on issues that are not socially sensitive can also be important for the team to discuss. For example, if there were disagreements on the question “client goals are clear,” then the team members for whom the goals are clear might wish to explain them to the people for whom the goals are not clear.

Picking the other choices lists the questions in different orders. Picking “Education” lists first those issues that it is most important for team members to know about in order to work together effectively. These are usually recommended processes, such as “write the plan down” or “identify areas of expertise.” Picking “Recommendations” lists first those issues that have the most useful recommendations. These are the usually recommendations that are simple and not obvious or that suggest multiple ways to address an issue. Examples are “designate someone as an expert” to encourage sharing of private information, or “create a common repository for team products” to help people know each others’ abilities and to track task progress.

Selecting “Impediments” or “Symptoms” filters the issues by whether they are impediments or symptoms, and orders them by their impact on causing the enabler to be an area of concern. Thus, the order of the list reflects both the weight of the issue in the knowledge base and the fraction of the team that believes the issue is true for the team. With these selections, the tool lists all impediments and symptoms, and not just the ones that half the team viewed. This option is analogous to the drill down available in individual mode, and helps the team to review the full list of impediments and symptoms

that might be relevant to their team. Finally, picking "agreement" lists the issues in order of the fraction of team members who agreed the issue was a problem. With that selection, all questions with red squares would be listed first, followed by questions with yellow squares, and then by questions with green squares.

The two columns "Agree" and "Disagree" order the questions by the number of team members who agreed or disagreed with the questions. This is not necessarily the same as ordering them by "Agreement" in the previous column, because not all team members see exactly the same questions. The column "Enablers" lists all the enablers that each question can impact. Because many impediments make it harder for team members to obtain knowledge in more than one area and because most symptoms are ambiguous and could reflect a problem in multiple areas, many issues list more than one enabler in this column. Enablers listed in black are those the team selected to be addressed on this depiction. Enablers listed in gray are the other enablers that the issue impacts. Finally, the Recommendations column lists one summary recommendation for the issue. Clicking on the recommendation generates the full list of recommendations for that issue. This list is the same as would be obtained in the tool's individual mode.

Recommendations

When the team accesses the recommendations for a particular issue, the tool lists different actions to take to address that issue. When the team reviews the recommendations for a different issue, they see a different list of recommended actions, some of which may also be recommended for the first issue. The sum of the individual actions on several lists could total 30 or 40 (or more!) items, some of which are shared on several lists, and accordingly are able to address several issues simultaneously. A team viewing these lists in sequence might not be able to easily identify these actions able to address several items simultaneously, and that therefore may be the most beneficial.

The Recommendations interface (Figure 17) addresses this issue. It helps the team identify the one or two things that would be the most helpful overall. The interface lists specific recommendations in the order of the number of questions each addresses. It also lists the different enablers that taking the action could improve.

This ordering of recommendations takes advantage of the fact that most of the specific items on a recommendation list for one issue also occur on the recommendation lists for other issues. In fact, the most popular items appear on more than a dozen of these lists. By counting the number of these lists that a recommendation is on, the tool can help identify the actions that "kill the most birds with the fewest stones" by addressing multiple issues simultaneously.

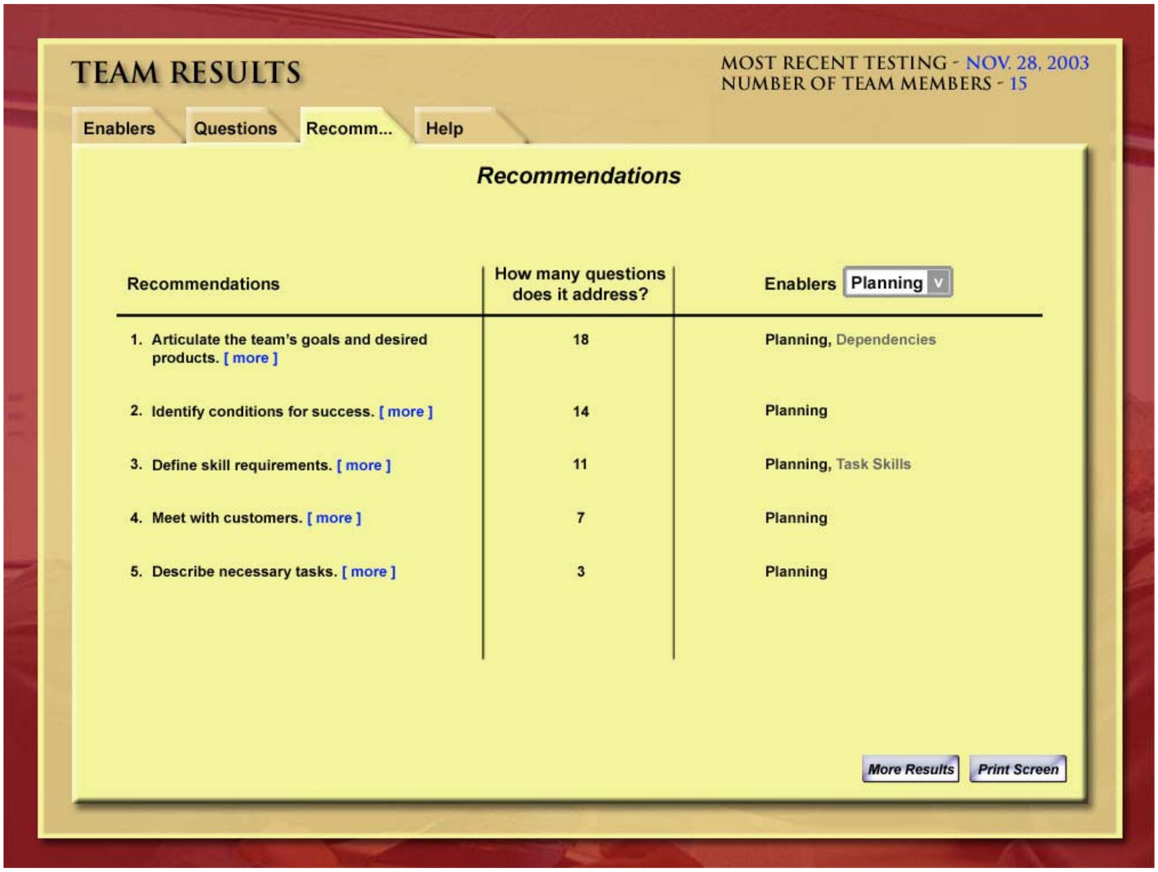


Figure 17. Team View Recommendations

3.1.4 Trends

Often teams use the Collaboration Advizor™ over several months, employing the tool every couple of weeks to assess their team's knowledge. Consequently, over time the team creates a sequence of team views, with each of these views being a snapshot of the team's knowledge strength at a different point in time. The team can access the any of these historical team views by specifying the view's date. Using the trends feature, the team can also look at the knowledge trends over time and see areas of improvement or backsliding.

Figure 18 is the main interface for reviewing trends. The left side is a list of enablers. The team can select any of these enablers to view changes in knowledge strength over time for that enabler. The team can also choose ask the Advizor™ Tool to select the enablers to view according to one of the three criteria: "view all enablers," "view the most improved," and "view the least improved."

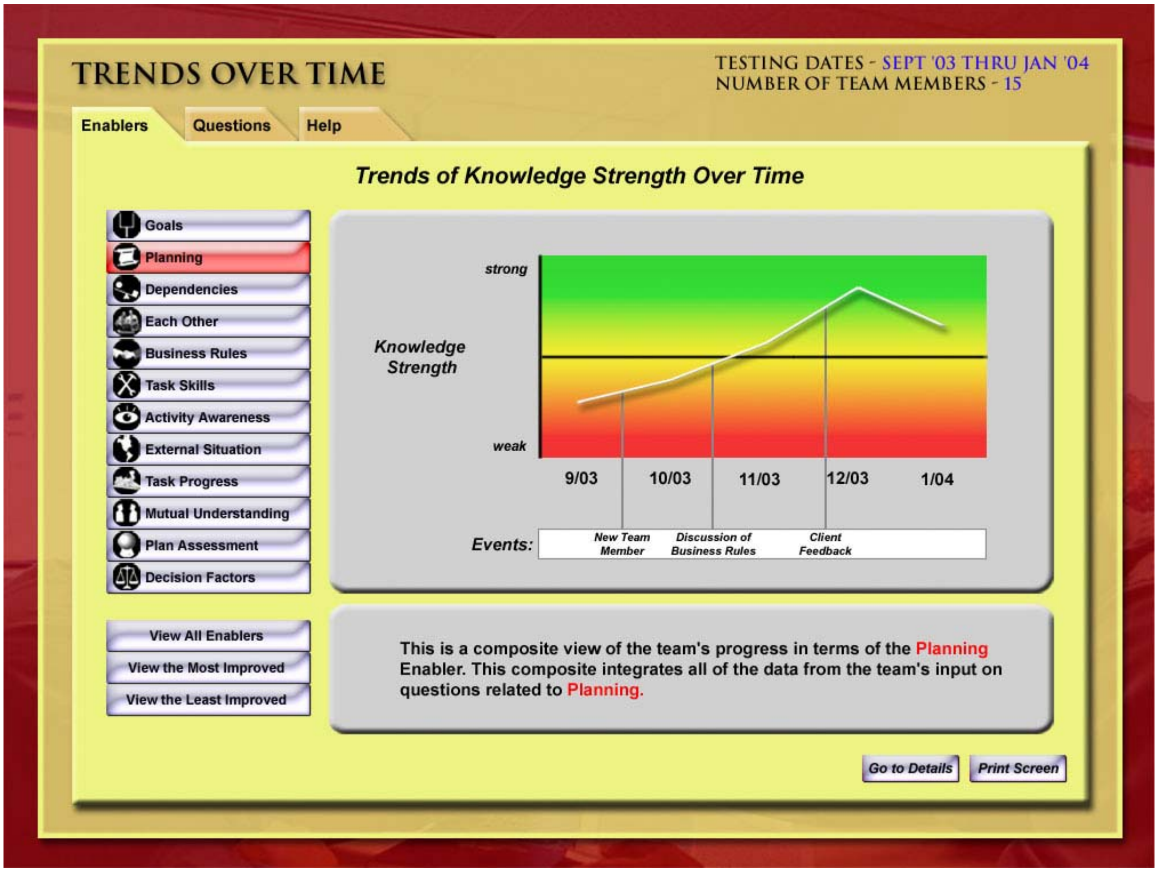


Figure 18. Enabler Trends

The right hand side of Figure 18 charts the change of knowledge strength for the selected enablers over time. It also overlays key team events, such as new team members, client feedback, or implementation of tool recommendations, that might have impacted knowledge strength.

Team members can also drill down on the trends and determine the issues that account for these changes. These are the issues that team members answered differently over time. For example, early in the team's work, many team members may feel that the client's goals are unclear. Later, as the team continues to discuss goals and receive client feedback, most of the team members may decide that these goals have been clarified. The change in this issue will therefore contribute to improved knowledge strength for the enabler "goals."

Figure 19 lists the specific issues which contributed to assessing knowledge strength at various points in time. This list groups issues by whether they improved (fraction of team members saying it is true of the team drops), stayed about the same, or got worse. The team can select particular issues to plot over time. Figure 19 plots three of these, issue #1, issue #4, and issue #6, and for each shows the percentage of team members agreeing that the issue is true for that team.

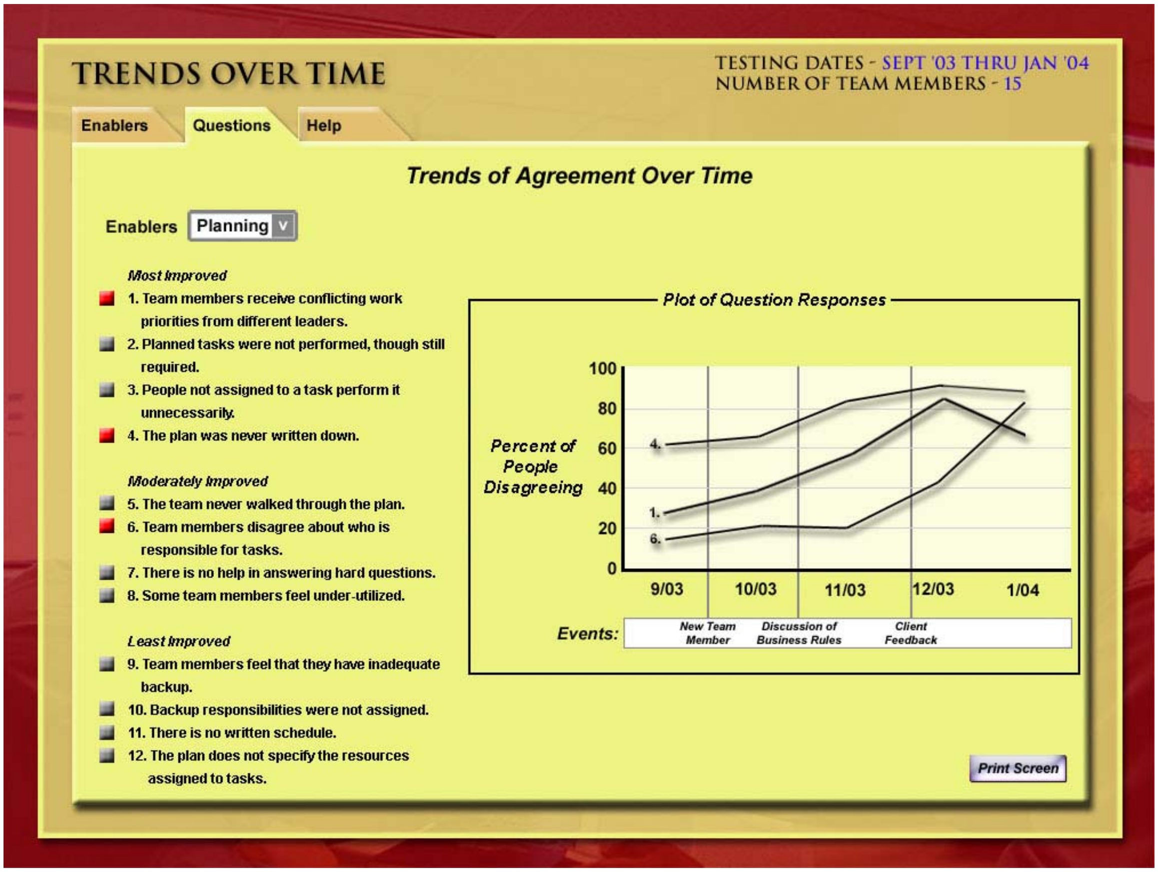


Figure 19. Question Trends

This trend view can help teams to assess not only how well they are doing cognitively, as each team view can do, but also helps them assess the effectiveness of their methods for improving their team. Thus it helps team improve their ability to improve.

3.2 Tool Knowledge Base and Algorithms

Section 3.1 described the tool from the user's perspective. It described the functions that the tool performs and the ways that individuals and teams interact with the tool. This section briefly describes how the tool works as a kind of "value driven expert system." It describes the tool's knowledge base and its processes for deciding what questions to ask, for diagnosing knowledge strength, and for creating the team view. This description is intended to help users understand how the tool works in order to use it more effectively. It is not intended to be an engineering documentation of the tool.

3.2.1 The Advizor™ Tool's Knowledge Base

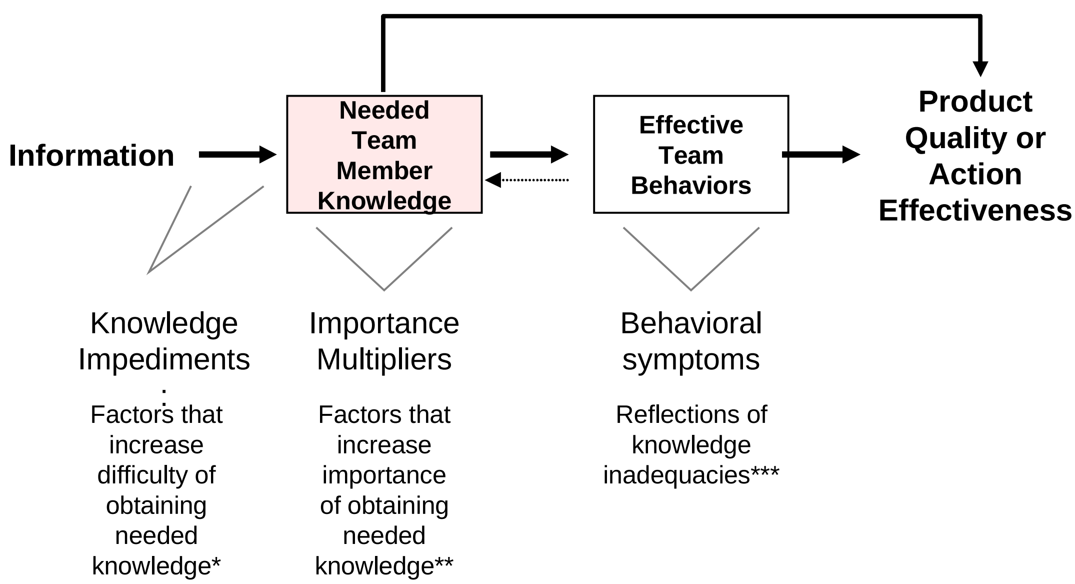
<u>The Tool's Perspective on Team Knowledge</u>

Impediments, symptoms, and importance multipliers. The Advizor™ Tool contains much of the knowledge described in Section 2.2.3 about what teams need to know in order to work together effectively, and it organizes this knowledge to help teams take

advantage of it. Figure 20 augments the previously shown Figure 2 to depict the different kinds of information that the tool uses to diagnose knowledge problems.

The top part of Figure 20 repeats the information-to-product trail in Figure 2. Here, team members obtain information which they must convert into a cognitive form—knowledge or understanding. This is the knowledge contained within the twelve enablers. This knowledge or understanding then guides team behaviors, which then, along with the knowledge itself, contributes to product quality or action effectiveness. Also as discussed, effective team behaviors contribute to needed team knowledge, since team members acquire some of the knowledge they need as a by-product of task performance.

The bottom half of Figure 20 depicts additional information that the Advizor™ Tool uses to help teams take advantage of its knowledge. This additional information includes knowledge impediments, importance amplifiers, and behavioral symptoms. Knowledge impediments are anything that makes it more difficult to acquire or have the needed team knowledge. The figure provides a few examples; e.g., cultural diversity impedes team members from knowing one another and geographical distribution impedes team members from knowing what each other is doing. The description for each knowledge enabler discussed impediments in the material on “obtaining the understandings needed for….[enabler name].” The Advizor™ Tool has somewhat more than 100 impediments, on the average about eight for each knowledge enabler.



*asynchronous, geographically distributed, cultural diversity, heterogeneous knowledge, unique roles, task dependencies, rotating team members

**highly interdependent tasks, poor management support, intelligent adversary, dynamic environment

***missed deadlines, failure to offer help, surprise at others behaviors, failure to share information, redundant or missed tasks

Figure 20. The Advizor™ Tool’s View of Knowledge

Importance multipliers are characteristics of the team, task, or situation that increases the importance of having a particular kind of knowledge. These are the conditions that increase the likelihood or severity of the consequences of knowledge shortfalls. For example, facing an intelligent adversary increases the importance of understanding the external situation and of understanding whether a plan will still work. Importance multipliers were discussed as part of “importance of …[enabler name].” The Knowledge Base has about 20 importance multipliers.

Behavioral symptoms are the overheard conversations and team behaviors that could indicate inadequate team knowledge in some area. They were discussed under the headings “evaluating the….” Unfortunately, team behaviors usually are not very diagnostic of the specific kind of knowledge shortfall responsible for the behavior. For example, the cause of “missed deadlines” might be a poor understanding of the team’s schedule, might be caused by overwork, or might be caused by poor task understanding. In addition, while the presence of a behavioral symptom may be excellent evidence that some knowledge problem exists, the absence of a symptom report usually is very weak evidence that the problem does not exist. That is because there are many reasons why a symptom may not be reported even though the team has a knowledge gap. For example, it may not have occurred because the team was lucky and no circumstance arose that would trigger it. Or it may have occurred but was not reported because no one on the team recognized its significance. The knowledge base contains about 100 symptoms.

Knowledge Base Components

The tool has an evidence knowledge base for diagnosing knowledge shortfalls, a recommendation knowledge base for making recommendations, and a question knowledge base for asking questions and listing issues.

Evidence knowledge base. The Advizor™ Tool’s knowledge base includes about 250 impediments, symptoms, and importance multipliers and associates each with the knowledge enablers that they are related to. For impediments, the evidence knowledge base specifies for each enabler how much each impediment hinders obtaining that kind of knowledge and how much the absence of that impediment facilitates acquiring the knowledge. For symptoms, the knowledge base specifies how much the presence of a symptom indicates a deficiency in each enabler category, and also specifies separately how much the absence of a symptom implies that the team’s knowledge is adequate in that area The importance multipliers do not contribute directly to a particular diagnosis. However, each impacts how much the tool will encourage the team to attend to possible problems in a knowledge area. Because there are 12 enablers and approximately 250 questions, the evidence knowledge base has about 6,000 entries that the tool draws on when it interprets individuals’ question responses to make a diagnosis.

Note that the knowledge base contains for each enabler/question pair (impediment, symptom, or importance multiplier) two numbers; a number to be used in the diagnosis if team members say the item applies to the team, and an independent number to be used if the team members says it does not apply. These two numbers enable the diagnosis to take advantage of both positive (when the item is checked) and negative information (when its

not). The weights to be used in the positive and negative conditions are independent, reflecting the fact that the conditional probabilities Prob(A | presence of something) and Prob(not A | absence of that thing) are independent.

Recommendation Knowledge Base. When a user asks for a recommendation about an impediment or symptom, the Advizor^{TM} creates the list of recommendations by drawing on a collection of about 125 actions. The Advizor^{TM} Knowledge Base contains these 125 actions, and for each impediment or symptom specifies which of these actions are to be included in the recommendation list and the order of the action on the list.

Question Importance Knowledge Base. The Advizor^{TM} uses this knowledge base to select the questions to be asked on the input forms and to decide the order in which to list them on the tool output displays. The Advizor^{TM} specifies the following information for each question: (1) whether it is an impediment, symptom, or importance multiplier; (2) whether it is asked on the leader set-up form; (3) whether it is appropriate to be asked when the team is still organizing; (4) whether it needs to be asked more than once over the life of a team; (5) its educational value; (6) its discussion value; (7) its recommendation value; (8) the enabler it is assigned to for question asking purposes; and (9) the priority for asking the question.

3.2.2 Advizor^{TM} Tool Algorithms

The Advizor^{TM} Tool algorithms select questions to be asked, record user answers, diagnose knowledge deficiencies, order and display results, consolidate team member responses and create a team view, and plot trends in knowledge strength and team responses to specific issues. Most of these algorithms are straight forward. Three that are not routine are the algorithms for question selection, knowledge diagnosis, and team view creation.

Question Selection

Intelligent question selection is a critical issue for the Collaboration Advizor^{TM} Tool. If the tool asks too few or unimportant questions, it will not receive the data it needs for a useful diagnosis. If the tool asks too many questions, teams will find using the tool to be too burdensome, and so will not use it. The tool knowledge base contains about 250 potential questions. Teams trying out the tool report that answering 50 questions is a reasonable workload.

The Advizor^{TM} Tool combines several different techniques to minimize the burden on the user while maximizing the value of the tool.

1. First, the tool off-loads some of the questions to the leader or team organizer. Prior to the team's using the tool, the leader answers preliminary questions about the nature of the team, task, and environment. These questions are all "importance enhancers" and all are factual (e.g., the team's work depends on the external situation). None of them have significant educational, discussion, or recommendation value. However, these leader questions help focus the tool on the knowledge areas likely to be most important to the team.

2. Second, the tool asks questions in multiple rounds, with the questions asked in the second round being influenced by the answers to the questions in the first round. The questions asked after the second round are optional and filtered by the tool's diagnoses.
3. Third, the tool skips questions that are inappropriate to the phase of the team's development. When the team is still organizing, the tool skips questions that are unimportant or not relevant to teams in this development phase. When the team has used the tool previously, it skips questions that have been asked before and whose answers are very unlikely to change over time.
4. To ensure that the most valuable and diagnostic questions for every enabler get asked in the first round, the tool "hard codes" the questions to be asked, subject to the team phase and re-asking constraints discussed in item 2. It asks two questions for each enabler.
5. To ensure that the most important questions are asked in the second round, the tool explicitly considers the questions' educational, discussion, and recommendation values, as encoded in the knowledge base. It also considers the extent to which a question is relevant to the enablers that are the most troublesome for the team, as judged from the answers to the questions in the first round.
6. After the second round, all questions are optional. To help focus the user on the most useful questions, the tool draws attention to those questions associated with enablers diagnosed as having the largest knowledge problems, lists the questions that have the highest educational, discussion, and recommendation values first, and separates the questions about impediments from the questions about symptoms.

Diagnosis in individual mode

When a team member uses the tool in individual mode, the tool asks that user about whether particular impediments, symptoms, or importance multipliers apply to that team. The tool then uses the user's answers and the question's diagnosis weights specified in the knowledge base to identify knowledge areas that need attention. As described previously, there are two of these weights for every question: a number to add to a concern score if the team member checks a question as being true for the team, and a separate independent number to subtract from the score if the team member does not check the item. The knowledge concern score is the sum of the weights of all checked items minus the sum of the weights of all asked questions that were not checked. The diagnosis algorithm does not consider unasked questions when making its diagnosis.

This linear scoring method and the use of approximate weights for diagnosis is a much less formal way of estimating knowledge adequacy than a "Bayesian" algorithm. However, the latter if done rigorously would require joint probability distributions for the probability of a knowledge inadequacy of some strength given a pattern of team member responses. Because the knowledge base size required for this would be enormous, because the baseline probabilities are unavailable anyway, and because the practical meaning of a knowledge inadequacy in terms of the likelihood of practical team problems is not defined, the Advizor™ Tool uses the much simpler linear method described above.

Although the diagnosis algorithm uses a linear diagnosis formula and highly approximate weights in the knowledge base, there are both theoretical and pragmatic reasons to believe that the diagnoses are accurate enough to be very helpful to teams. These reasons include:

1. The diagnosis does not require highly precise evidence weights for the diagnosis to be very helpful. The purpose of the diagnosis to help the team identify potential problem areas that they should discuss. Approximate answers are accurate enough to do this.
2. This expert system is “value driven.” It works by looking at the preponderance of risks and symptoms in a particular area. Value driven expert systems give good results if they address all of the driving factors. So long as they do this, the exact weights assigned to one factor do not matter very much.
3. If the tool fails to identify a problem area at some point in time, then the problem is unlikely to be very serious. Should it become more serious, then the tool will catch the problem at a later date.
4. Teams that have used the tool report that it provides good, useful results appropriate to their team.

<u>Creation of the team view</u>

The team view interface is the focal point of the team meeting where members review how the team is doing, identify possible weaknesses, air differences in viewpoints, and decide how to fix problems. The interface needs to help teams focus on the most important issues. The tool’s algorithms for diagnosis and for prioritizing issues support this focusing.

Generation of team diagnosis. This algorithm has three steps: collection of response data, generation of response statistics, and enabler score calculation.

When individual team members exit from the Advizor™ individual mode, the tool records which questions were asked and the team members’ responses to these questions. To prepare for the team diagnosis, the tool gathers these responses from all team members and then calculates for each question the number of people who were asked the question and of those, the fraction that said it was true of the team. The tool then discards from further consideration all questions that fewer than half the team members saw. Using the remaining questions, it calculates the enabler diagnosis scores using a slight modification of the formula employed in individual mode. For each question that more than half the team was asked, it adds to the score:

weight if question is true of team*fraction of team that felt it was true of the team –
weight if question is not true of the team*fraction that said it was not true

The fraction of the team in the formula above is the fraction of the team who were asked the question. The diagnosis score is then the sum over all questions that at least half of the team were asked.

Prioritization of issues. Once its diagnosis calculations are completed, the Advizor^{TM} generates the team view display. It displays the enabler strength scores calculated above using the bar charts of Figure 15. It then prioritizes the individual impediments and symptoms so that it can list the most important issues first.

This priority for an issue depends both on its static value to the list ordering criteria (e.g., education, discussion, or recommendation) and also to the extent to which it is contributing to the low scores of enablers with low scores. The static value is extracted directly from the Question Importance Knowledge Base, with the value used depending on the ordering criteria. The extent to which it is contributing to low score enablers is the maximum over enablers of:

enabler's current score * weight to enabler score if item is true* fraction who said it's true of team

Thus, an item's priority for listing on the team view output increases as the level of concern for the enablers it impacts increases, as the extent to which the item impacts that score increases, and as the number of people who agree that the issue is true of their team increases. Items that impact only enablers that are not of current concern have low priorities, as do items that the team feels are not true of the team or that do not have much of an impact on any enabler of current concern.

3.3 Evolution and validation of the tool

The Advizor^{TM} Tool organizes the knowledge which the theory of Section 2 proposes as the cognitive foundation for effective teamwork. It also helps teams use this knowledge to improve their performance. This section reviews the evolution of the tool, with special focus on the identification of problems encountered, on methods to overcome these problems, and on work to validate the tool correctness and utility.

The paper guideline phase: motivation for computer-based tool

Before EBR developed the Collaboration Advizor^{TM} Tool, it created paper-based guidelines to help teams diagnose and fix knowledge shortfalls. These guidelines were based on:

1. Theories and models for how team members use knowledge to work together effectively. These were drawn from research papers on collaboration. They are documented in EBR's Phase 1 report, "Metrics for Evaluation of Cognitive Architecture-Based Collaboration Tools" (Noble, 2000). That work was the precursor to the material in Section 2.
2. Advice on how to improve collaboration, particularly books in the popular press (Brounstein, 2002; Maxwell, Herbelin, 2000, Katzenbach and Smith, 2001; Katzanbach and Smith, 1993)
3. Twenty case studies of team failures. EBR's subcontractor, Klein Associates, provided these examples from its research files. Section 1 illustrated two of these cases. EBR categorized these 20 cases in terms of the underlying cognitive reasons for the shortfall, and identified nine basic cognitive reasons for the failure.

These nine categories have since been incorporated into the twelve knowledge enablers.

EBR tested the paper guidelines on two different teams: the EBR war room team and a JFCOM evaluation team. Several members from each team filled out paper forms answering questions about observed impediments and symptoms for their teams. EBR then transferred these answers to an Excel spreadsheet, and computed the correlation between a team being at risk for a cognitive problem and a team showing symptoms of having that problem. In both cases, impediments and symptoms correlated positively. These results provided preliminary evidence for the underlying concept of our theory: that there are fundamental knowledge categories important for effective teamwork, that knowledge impediments would reduce the adequacy of knowledge in an area, and that this inadequacy would be reflected by behavioral symptoms.

EBR asked six people to review the guideline book, three paid under the contract and three volunteers. When the volunteers did not do so, EBR decided to create a better delivery vehicle than paper for helping teams understand and apply the information in the guidelines. Because in the guideline tryout EBR had transferred team member responses to Excel to analyze results, we decided to build a computer-based guideline system on top of that application. With this system, team member responses would automatically be encoded on a spreadsheet convenient for analyzing the data.

3.3.1 First tool version: individual mode

The first Excel version of the tool was a computer-hosted version of the paper guidelines. Individuals filled out the same forms on the computer as they did on paper. The computer then recorded their responses, and calculated a knowledge score for each of the enablers. This tool was a preliminary version of the current tool's individual mode. It illustrated the concept for the Collaboration Advizor™. It provided a hands-on vehicle for people to try out the guidelines.

In this tool phase, EBR received further indications that the tool concept was valid. Many people tried the tool, and most thought that its diagnoses were accurate. EBR discovered from these initial tool evaluations that the tool gave good responses across a broad range of cases, including the EBR war room, JFCOM experimentation teams, NATO research panels, and even a domestic team trying to solve a family problem. Though in general users endorsed the tool's performance, there were some cases where a user did not think the tool's diagnoses were completely on target.

EBR's subcontractor, Klein Associates, tested the tool's applicability for several of the teams that they studied in the past. In some cases, they regarded the tool's assessments to be highly accurate. In other cases, they felt it overlooked significant issues. As these issues were documented, EBR upgraded the tool's knowledge base, improving its performance so that in those cases the tool was judged to perform well.

During this phase, individuals from EBR's war room team continued to use the tool. After doing so, team members met to discuss their experiences. EBR noted that these

discussions could be more productive if the tool could collect and display the individual team responses.

3.3.2 Second tool version: team view

The second version introduced the tool's "team view" described in Section 3.1.3. During this phase, the EBR war room team exercised the tool three times and a second EBR team, the Command and Control Research Team, used it once.

For each of these trials, team members used the tool's individual mode sequentially. After all team members responded to the tool's questions, EBR created the team view showing an overall team knowledge strength diagnosis, and listing the significant questions that had been asked. The team then met to discuss results.

In this phase, EBR observed how the tool supported key team dynamics. First, it was observed that the team view often became a launching point for team discussions about critical issues, some of which had not been explicitly addressed by any tool question. For example, in one case the team discussion identified central team problems to be the requirement for team members to respond quickly to impromptu task requests from clients and by the requirement to respond to tasking from multiple leaders. Neither of these was directly raised by the tool. Second, it was observed that the tool enabled the team to surface and discuss critical issues that might otherwise not have been raised. Because the individual input was anonymous, the tool gave everyone an equal voice, whether the member had just joined the team or was its leader. Third, because it was the tool rather than an individual that raised issues, topics that might have been socially awkward to introduce without the tool could now be discussed without breaking any norms of polite behavior. Thus, the team could now discuss such issues as "not everyone is adequately trained to carry out his tasks," or "some people do not understand appropriate team behavior." Fourth, because the team view provides statistics on how many team members agreed or disagreed with what particular issues were true of the team, the tool helped team members understand areas of agreement or disagreement. For instance, the tool could point out that half the team thought that team goals were clear while half thought they were not. Fifth, it became apparent that the tool was educational about processes teams usually need to follow in order to be effective. Sixth, the tool helped the team's leader understand the viewpoints of team members. There were many occasions where the leader was surprised by the views of the team.

The war room team used the tool three times over a 2-month period. After each of the first two uses, the team identified and carried out tool recommendations. Team members felt that doing this helped the team address important issues.

This version of the tool was the one that EBR demonstrated at the Navy Opportunity Forum in May 2003. This tool generated considerable interest at that trade show. The organizers told EBR that our briefing was the best attended. We demonstrated the tool to dozens of people at the show, and all agreed that it was a unique prospective on collaboration that they had not seen before.

However, as the war room team continued to use the tool, we became concerned that the tool was not always addressing the most significant issues. Between the second and third times that the war room team used the tool, that team discovered that some of its members had very strong differences over the team's business rules, and discovered that they had not developed good procedures for resolving these differences. When the tool failed to reflect these new symptoms of poor understanding of business rules, EBR reviewed our impediment and symptom lists and the tool algorithms for selecting questions to ask. With extensive input from the war room team and a renewed review of the collaboration literature, EBR created a more comprehensive list of critical knowledge "sub-sub-enablers." Section 2.2.3 documents these details knowledge requirements in its description of knowledge requirements. EBR then adjusted the impediments and symptoms in the tools evidence knowledge base to ensure that the tool could consider the full span of this more comprehensive list of critical knowledge.

During this tool review, EBR also tested the tool's question selection algorithms. These tests showed that these selection algorithms did a poor job of picking those questions that the team members believed were true of their team. To address this problem, EBR modified the question selection algorithm to the one described above in Section 3.2.2.

3.3.3 Third tool version: individual issue exploration

This version of the tool incorporated the improved list of symptoms and risks and the upgraded question selection algorithm. It is the version tested at the Naval Postgraduate School in August and September 2003.

To prepare for these tests, EBR reviewed the Collaboration Advizor™ Tool with the faculty member who teaches collaboration infrastructure and whose class would provide the test environment. He pointed out the critical role that the tool plays in a team's development process. In his view, the Advizor™ Tool provides the "control signal" that teems need in order to know that they need to adjust and to be aware of the specifics of what needs adjusting.

This collaboration class used the tool twice, once in August and once in September. This tryout was a significant milestone in the tool's development, for it was the first time that a disinterested (non-EBR) team assessed the tool. After the first use, the tool diagnosed a few areas where team knowledge could be weak. In the team view discussions, team members concurred that this was accurate. Three weeks later, the team used the tool a second time. This time, the tool diagnosed no areas where the team was weak. In reviewing these results with the team, team members said that they had discussed the issues the tool raised on the first occasion and addressed them. They felt that the tool contributed to their identifying and addressing some critical team issues.

In reviewing the team views for the first and second tool uses, EBR identified those specific questions that accounted for the change in the tool's diagnosis. These were the questions that team members felt were true for their team the first time they used the tool and that were no longer true the second time they used it. This tracking of team changes

and analysis of the reasons for the change are incorporated in the "trends" functions of the commercial version of the tool.

As part of its participation at the NPS, EBR was a guest lecturer at one of the classes, explaining the cognitive foundations for effective collaboration and how the Collaboration Advizor™ Tool works. In these discussions, EBR related the twelve knowledge enablers to the normal military planning and decision processes, pointing out that these enablers are natural extensions to the commonly taught "OODA" (Observe-Orient-Decide-Act) cycle. In discussing the tool's recommendations, we pointed out that they draw on four important sources: (1) popular literature on how to help people work together better; (2) program management; (3) command and control theories and military standard procedures; and (4) theories of situation assessment and decisionmaking.

ONR had hoped that the Naval Postgraduate School evaluations could be formal and objective tests of the tool's accuracy and utility. ONR had desired statistically significant results showing the extent to which teams that used the tool had better collaboration results than did teams who did not use the tool. Such tests were not possible at the postgraduate school because of the small number of students in the collaboration class. Accordingly, ONR searched for another venue for a statistically controlled evaluation of the tool.

NAVAIR agreed to help ONR carry out these tests, permitting the Advizor™ Tool to participate in its planned collaboration experiments. These tests are planned for spring 2004.

In order to achieve the numbers required for statistical control, the NAVAIR experiments use teams that meet for only a few hours and use an abbreviated form of the tool. Thus, the tests cannot address many of the hypothesized benefits from the tool. For example, they cannot show how the tool helps teams adjust over time because the teams meet only once. However, the tests can measure other hypothesized tool benefits, such as the tool's educational ability. Thus, the NAVAIR evaluation can test the following hypothesis: teams whose members use the tool understand critical collaboration issues better than teams that do not use the tool, and teams that understand these critical issues better create a better team product.

The scheduling of the NAVAIR tests prompted a review of the educational value of each impediment and symptom in the tool knowledge base. EBR revised the impediment and symptom lists in the tool after this review. Because of the short time that the NAVAIR teams would meet, EBR added the "leader" questions to save subject time when using the tool. All of these changes became part of the general tool to be evaluated by teams in the United Kingdom.

3.3.4 Fourth tool version: focused questions

This was the final prototype developed using Excel. It included all of the knowledge base and question selection upgrades discussed so far. This was the version "beta-tested" by two teams at Dstl in the United Kingdom.

Given the encouraging reception of the tool by numerous types of teams, ONR provided funding to develop a "commercial" version of the tool independent of Excel and suitable for use in a Web environment. Also at this time, EBR submitted a patent application for the tool.

As a first step toward developing the commercial version, EBR developed a set of story boards to improve the tool's interface. Several of these are depicted in Figures 12 through 17.

In December 2003, EBR reviewed the tool and the proposed interface design with the two teams in the UK. Feedback from these teams was particularly significant, since their members had extensive human factors and collaboration backgrounds. Both teams reported the tool to be useful. Both teams agreed that the tool made a useful contribution to team functioning, noting it can support such diverse team types as task teams formed to create a particular product, and resource teams formed to provide general support within an organization.

Both teams made helpful suggestions for using the tool itself and for improving the interface. Much of the discussion however focused on how the tool should handle leadership and motivation, two important non-cognitive issues that the tool does not address in detail. To prepare for augmenting the tool's treatment of motivation, EBR created a draft "motivation" enabler, which the UK teams reviewed and generally accepted. It was pointed out, however, that the motivation and leadership issues would need to be handled with great sensitivity. It was felt that many leaders would avoid any tool that might diagnose their leadership to be ineffective. Furthermore, because poor motivation on a team is often the result of poor leadership, team leaders might also avoid any tool that documented poor team motivation.

The importance of leader sensitivities was highlighted in the UK when one of the three teams scheduled to participate in the beta-testing dropped out. Part of the reason for this was that the team leader recognized that his team had some problems that he was planning to address. This leader felt that the tool might interfere with his chosen method of addressing team difficulties.

Commercial version

The commercial version of the tool is scheduled for completion in summer 2004. It will incorporate the full functionality described in this report.

References

Argote, L. and P. Ingram. (2000), Knowledge Transfer: A Basis for Competitive Advantage in Firms. Organizational Behavior and Human Decision Processes, Vol. 82, No. 1, (pp 150-169)

Beach and Mitchel, (1987). Image Theory: Principles, goals, and plans in decision making. ACTA Psychologica, Vol. 66, No 3.

Brounstein, Marty (2002). Managing Teams for Dummies. Wiley Publishing, Inc., Indianapolis, Indiana

Canon-Bowers, J.A. Salas, and S. A. Converse. (1993). Shared Mental Models in Expert System Team Decision-Making. In Individual and Group Decision Making: Current Issues, eds. M. J. Castellan. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc./

Evidence Based Research (2001), *Enabling Effective Collaboration in Military Operations. Workshop Report.* Evidence Based Research, Vienna, VA.

Evidence Based Research (2002], Collaboration Guidelines, Identifying and Fixing Collaboration Problems. Draft. Evidence Based Research, Vienna, VA. Katzenbach, J.R. and Smith D.K. (1993); The Discipline of Teams. Harvard Business Review, 71 (2): 111-120

Graetz, K., Boyle, E., Kimble, C., Thompson, P., Garloch, J. "Information sharing in face-to-face, teleconferencing, and electronic chat groups," *Small Group Research,* V29(6), 1998

Herbelin, Steve. 2000. Work Team Coaching. Riverbank Book, Riverbank, CA.

Hollenbeck, J., Ilgen, D., Sego, D., Hedlund, J., Major, D., Phillips, J. (1995). Multilevel theory of team decision making: Decision performance in teams incorporating distributed expertise. Journal of Applied Psychology, V. 80 (2).

Janis, Irving L. *Victims of Groupthink*, Houghton Mifflin Company, Boston, 1972.

Katzenbach, Jon R. and Smith, Douglas K. 1993. The Wisdom of Teams. HarperCollins, New York. NY.

Katzenbach, Jon R. and Smith, Douglas K. 2001. The Discipline of Teams. John Wiley and Sons. New York. NY.

Kirkman, Bradley and Debra Shapiro. 1997. "The Impact of Cultural Values on Employee Resistance to Teams: Toward a Model of Globalized Self-Managing Work Team Effectiveness." Academy of Management Review: 22

Kirzl, J, Noble, D., and Leedom, D. (2003); Command Performance Assessment System. Evidence Based Research. Vienna, VA.

Klein, Gary. 1998. *Sources of Power: How People Make Decisions*. The MIT Press. Cambridge, Massachusetts

Levine, J., & Moreland, R. *Progress in small group research. Annual Review in Psychology*, V 41, 1990

Liang, D., R. Moreland, and L. Argote. "Group Versus Individual Training and Group Performance: The Mediating Role of Transactive Memory," *Personality and Social Psychology Bulletin*, Vol. 21, No. 4, 1995, pp 384-393.

Mathieu, J, Goodwin, G, Heffner, T, Salas, E, and Cannon-Bowers, J. (2000). The Influence of Shared Mental Models on Team Process and Performance. J. of Applied Psychology, Vol. 85. No. 2 (273-283)

Maxwell, John C. 2001. The 17 Indisputable laws of Teamwork. Thomas Nelson Publishers, Nashville.

Noble, D. Truelove, J. Grosz, C. and Boehm-Davis, D. 1989. A theory of information presentation for distributed decision making. Engineering Research Associates report, Vienna, VA.

Noble, D. (2000): "Metrics for Evaluation of Cognitive Architecture-Based Collaboration Tools." Evidence Based Research. Vienna, VA.

Noble, D. and Letsky, M. (2002). "Cognitive-Based Metrics to Evaluate Collaboration Effectiveness," SAS symposium "Analysis of Military Effectiveness of Future C2 Concepts and Systems," SAS 039 Symposium, Den Hague, Netherlands

Noble, D. "A Cognitive Description of Collaboration and Coordination to Help Teams Identify and Fix Problems," in *Proceedings of the 2002 International Command and Control Research Technology Symposium*, Quebec, Canada, 2002.

Noble, D. and Kirzl, J. (2003); "Objective Metrics for Evaluation of Collaborating Teams." Proceedings of the 2003 Command and Control Research Technology Symposium. Washington, D.C. National Defense University

Noble, D. (2003). "Understanding and Applying the Cognitive Foundation of Effective Collaboration," in Proceedings of the 15th international Conference on: Systems Research, Informatics and Cybernetics: Collaborative Decision-Support Systems Focus Symposium, Baden-Baden, Germany

Noble, D. "Understanding and Applying the Cognitive Foundation of Effective Collaboration". in *Proceedings of the 5th ONR Conference on Collaboration*, Quantico, VA, 2003.

Noble, D. (2004). A Knowledge Theory of Collaboration Tools. Paper submitted to IEEE Expanding the Boundaries of E-Collaboration: A Special Issue of the Journal: IEEE Transactions on Professional Communication

Noble, D, Shaker, S, and Letsky, M. 2004. "The Cognitive Path to Team Success." Government Executive. In press.

Noble, D. 2004. Knowledge Foundations of Effective Collaboration, in *Proceedings of the 2004 International Command and Control Research Technology Symposium*, Copenhagen, Denmark, 2004.

Salas, E. Dickenson, T. L., Converse, S. A., and Tannenbaum, S. I (1992). Toward an understanding of team performance and training. In R. W. Swezey and E. Salas (Eds.), *Teams: Their training and performance* (pp. 3-29). Norwood, NJ: Ablex.

Tuckman, B. W. 1965. Developmental Sequence in Small Groups. Psychological Bulletin, 63. pp. 384-399.

Wegner, D.M. (1987) Transactive Memory: A Contemporary Analysis of Group Mind." In Theories of Group Behavior, eds. Brian Mullen and George R. Goethals, 185-206. New York: Springer-Verlag.

Zsambok, C. E., Klein, G., Kyne, M., and Klinger, D.W.(1993); Advanced Team Decision Making: A Model for High Performance Teams. Fairborn, OH: Klein Associates, Inc.